Simplify your Code with Helmfile

ROOT LEVEL TECHNOLOGY

BRADEN WRIGHT | ROOT LEVEL TECHNOLOGY

codefresh

Braden Wright

Staff Cloud Architect

ROOT LEVEL TECHNOLOGY
Your cloud strategy partner.

braden.wright@rootleveltech.com

Agenda

What's not covered?

- Kubernetes 101
- Helm 101
- Codefresh 101

https://github.com/rootleveltech/codefresh-helmfile-webinar

Agenda

Common Problems Helmfile can help with...

- Why Helmfile
- What is Helmfile
- Helm Repos
- More Hooks
- Helm Diff
- Another Approach to Secrets
- Helm Dependencies
- Multiple Environments
- Templating Values yaml

https://github.com/rootleveltech/codefresh-helmfile-webinar

Why Helmfile?
I love Helm but if only I could ...

- Setup Helm Repos
- Automate something to run before / after Helm
- See what's going to change before Deploying
- Manage Secrets as part of Helm Deploy
- Have dependencies installed as separate Helm Deploy
- Control order of my Helm chart dependencies
- Template values.yaml
- Get data from another source / run a script
- Use Environment Variables

What is Helmfile?

- Wrapper on top of Helm
- No Vendor Lock-in
- Golang Templating without Restrictions

https://github.com/roboll/helmfile

What is Helmfile?

```
$ cat helmfile.yaml
repositories:
- name: stable
  url: https://kubernetes-charts.storage.googleapis.com

releases:
  - name: myapp
    namespace: default
    chart: ../../charts/myapp
    labels:
      name: myapp
    version: 0.1.1
  - name: dns
    namespace: default
    chart: stable/external-dns
    labels:
      name: dns
    version: 2.20.4
    values:
      - values.yaml.gotmpl
```

Demo Disclaimer

- Walkthrough deploys
- Starting from a completely clean GKE install
- Setup in GCP for things like DNS, credentials, etc.
- Mono Repo for our example

Helm missing Repos

```
$ helm upgrade --install mytraefik stable/traefik
Error: failed to download "stable/traefik" (hint: running `helm repo update` may help)
```

```
repositories:
- name: stable
  url: https://kubernetes-charts.storage.googleapis.com
```

Helm missing Repos

```
$ helmfile apply
Adding repo stable https://kubernetes-charts.storage.googleapis.com
"stable" has been added to your repositories

Updating repo
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "stable" chart repository
Update Complete. ⎈ Happy Helming!⎈
```

Helmfile Hooks

Events: prepare, presync, postsync, cleanup

```
$ helm upgrade --install myapp . --namespace my-namespace
Release "myapp" does not exist. Installing it now.
Error: create: failed to create: namespaces "my-namespace" not found
```

Helmfile Hooks

Example: create Namespace before deploying chart

```
$ cat helmfile.yaml
releases:
- name: myapp                    # name of this release
  namespace: default             # target namespace
  chart: ../../charts/myapp      # the chart being installed to
  version: 0.1.1
  hooks:
  - events: ["prepare"]          # Supported Events: prepare, pr
    showlogs: true
    command: "../../scripts/create_namespace.sh"
    args: ["dev"]
```

Helmfile Hooks

Example: create Namespace before deploying chart

```
$ helmfile apply
Building dependency release=myapp, chart=../../charts/myapp
helmfile.yaml: basePath=.

hook[prepare] logs | Missing Namespace dev creating now
hook[prepare] logs | namespace/dev created
hook[prepare] logs |
Comparing release=myapp, chart=../../charts/myapp
********************

        Release was not present in Helm.  Diff will show entire contents as new.

********************
default, myapp, Deployment (apps) has been added:
-
+ # Source: myapp/templates/deployment.yaml
```

Helm Diff

- Want Terraform Plan for Helm?
- Bug for Helm 3, 3-way diff

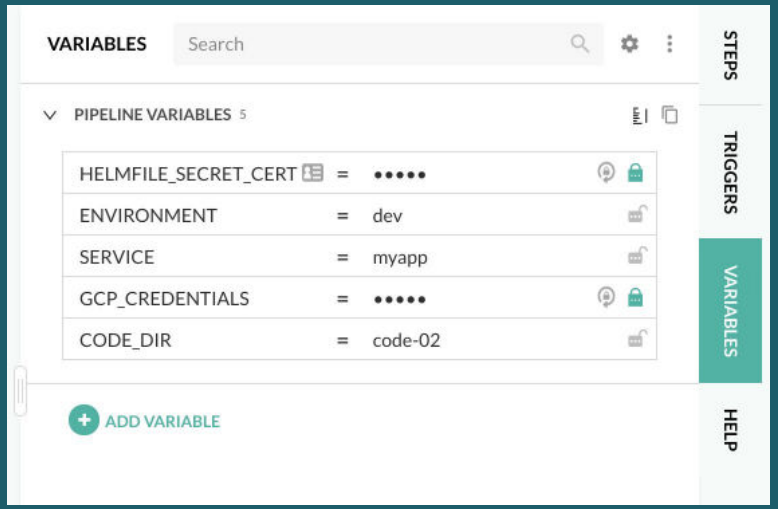
```
      app.kubernetes.io/instance: myapp
default, myapp, Deployment (apps) has changed:
  # Source: myapp/templates/deployment.yaml
  apiVersion: apps/v1
  kind: Deployment
  metadata:
    name: myapp
    labels:
-     helm.sh/chart: myapp-0.1.1
+     helm.sh/chart: myapp-0.1.0
      app.kubernetes.io/name: myapp
      app.kubernetes.io/instance: myapp
      app.kubernetes.io/version: "1.16.0"
      app.kubernetes.io/managed-by: Helm
  spec:
    replicas: 1
    selector:
      matchLabels:
        app.kubernetes.io/name: myapp
        app.kubernetes.io/instance: myapp
    template:
      metadata:
        labels:
          app.kubernetes.io/name: myapp
          app.kubernetes.io/instance: myapp
      spec:
        serviceAccountName: myapp
        securityContext:
          {}
        containers:
          - name: myapp
            securityContext:
              {}
-           image: "nginx:1.16.0"
+           image: "nginx-broken:1.16.0"
            imagePullPolicy: IfNotPresent
            ports:
```

Helm Secrets

- How do you create secrets?
- Where do you store sensitive data?
- How do you distribute / share sensitive data?

https://github.com/zendesk/helm-secrets

Helm Secrets




```
$ cat secrets.yaml
secretFiles:
    gcloud-credentials.json: ENC[AES256_GCM,data:uUumBvNxQSE5KcnFnZV9HOgjw35+ZTTfP3sX
wTkG57sS76fLKqg1OYueZcCqWrGcZq95Tm0pa5zSuOXf8UIK1urV5uSJR6zQPBr8v15y020VntbQUe00JRGhr
IaAQ88dNt1YgCg/KL2RjdrFGJUlcjkQI5QPitZ0vWHhuagyHtulqgwZg+Q45mmMsxI+iGoyaNs2NVLt8CtZRN
znHd7U5wBrkysYfeVqqR9FUMAcwaVkH4KXcL3keSFxZSi+DUpUVBdXSkIDYp29h4B3lALag8f7R9QAB5leR/R
n8A1g5PrsQEgCNU0Y/30ZR2vlphl3NM4VLToCxXBYv6/A42AxA/HOr5jls22zPeffksh2AQBHiYH5O27ORjy4
uOH1SF64+DrJ6UNS7ZnudCubRflnBye6C3YyF7Yyo5G3Yr6Q7qYjCVGrk1VPtzrf3zmN6SgzZCTZ9nQLjRz9s
yHTE0ngIdY0r2xmZnL/AW7wUdg1b0lVnrFSRBMJTrOjx3jEmVE2MMeCx0u+/eSUCUWne4yF6FECJAVae1uTMF
ep5KhEPj4PjN8C2+1MXRHgrBK1nsWfUIHZwvBrAMQD9m49QlX6VwmRI1D03U4LYtszCfjKPqaXAHkWOxsGwiJ
+RIj3N5ZukNAVwdOYyuEV6u9Ei0FzLRQivRi/WkEcnzRGE7PwnCPjLvBPLhx/z9ftGeba4qjIp++9msb/dUF2
VzwMIOqlQO+3b/Fn0hSnIDgBIvysEHNX1o0p/YyORaGhDBUa9xEVHYGU5bMWlPP3uEiAff0KzVU+BFvcitMz6
EPI6URP0QR1dulDbXDNP4TgbTXb68iQSeroWSnStgfhThEmrn19LFQhyDoB/Sx3zwgcDJTziRdiqav0tGap4c
AW6TvIx6mI3+BQ8wA8MLvPIkCTN45O1q9uKAt+KtbnfuFeAUYVmAuO7+J1djRmGdgjt2kmecfNn+wJjH+typY
QAuFw9qJld65YdtkdZJy3HaDzhrh5iIYHF1Jdjgsttan2c1teq/IL3H5lMDVmOOuSZ5FjhAHpl36n9Fyd+DSX
ECX1g0+XtPc3BhlUppBYLwIVQbEWe15AZ3ZktDDHTC9vMaWZkZeaGGJEm5DORLLOyzQeGLHdwuBGPtEy4JC69
j8nMJBLolp6iyzv5uKbBwQWd5zZFeLLUWP17JKx1nD3T2x4J2YPuKAJ96IwEiSH2Hqn7KCaZm2sjGTy7pkWRy
a3XYxoS/SqWcywH3aS792+z/8H84n4MIArOI3QEg5SwAnvVgilT/cBj3hALaqdByw2NKtxmSvuB8ElkJHPu7Q
zRBScOeLhJppjkdCl5VCvsFD2xwWyPy+zYw4xoOQPgKKCMGc7x94Tn24fcnlnQfoOY7coDM24yU9f1sZ+DOmAC
K2cc66ibvntf/Lvh9L9tYtYAfGClK7l8JhJwyppeQhUE6+uyW5DDH4o10IvKCHe5PKa9u5davFewsTpeU1K3vt
YjSMjA6odQewzPNDo4/lKX36yZdDkVSmBSUnNOCRPSNjkWtf/qGKomdASAPLdOcH+f3HWRbocK8+U/WVj/Vvb7
aeQ=,tag:dRDyOGpbkZwOg+qqe+wKwQ==,type:str]
sops:
    kms: []
    gcp_kms: []
    azure_kv: []
    lastmodified: '2020-04-09T02:14:21Z'
    mac: ENC[AES256_GCM,data:mojtdull3D8/Y6YvRLwGUkduwhP1Lgwcv6T1Df3iIYgTdYAU+Hs3KHSS
owjjxyXm5/EtTTV9pt9QTab+XKXBzhUF8=,iv:Lh5Cn7oJhjg6kd4928dsk3NO3BPsmwPdEtJw1JLSUbI=,tag
    pgp:
    -   created_at: '2020-04-09T02:14:20Z'
        enc: |
            -----BEGIN PGP MESSAGE-----

            hQEMA4NvoX2CBO+rAQf7B+bZg4T3U9+GgkhqWHP/lUyHzDlRPBQGmZllNsfxarpW
            MwZxQHDEJVn1HbA8GHkawgy8QftNGO/IFsKnixGiSdgxjR04P8a8R+zfUdgz232Y
            PLj9wmAfteytmobSWJyTq2anK7rpLeUYFq/iQfTroFOUA8SuKqFkHYqeSCL2J4qm
            2hME7JsrzrtwYDO4nNSfr5wKaDe5KJyzi6dXo6a2zAhx5MzOinzRK5s3vEbeMtLv
            niMTc4vajdKrMoLeNcEWg/CM4NpHEzU/Gafc/uFaXL43wARUtX4aMhlpoxkdarM0
            e2735xL2WKjzOD4BWD+RxPl37b01f3C/H0g5htgt5tJeAYhUvXFZFiggSwnwPXgL
            v74KAt5stKyHm6vMCTDZ6QRhuc1V4Lg3L5LIcUjeVoyMOBdx+t5xp5u0Z5oEq1YJ
            AEPw6BAB8wzYXqVnoqOwKcBcc5nz7Gw6PFA8kLDZUA==
            =kOqM
            -----END PGP MESSAGE-----
        fp: A92179D1210A37F85CE7FA08572EB2D49A51C9CA
    unencrypted_suffix: _unencrypted
    version: 3.5.0
```

Helm Secrets

```
$ helm secrets view secrets.yaml
secretFiles:
    gcloud-credentials.json: '{ "type": "service_account", "project_id": "witronix-sandbox",
        "private_key_id": "9cf4e259331cd9c574e36ace34e393b2319acd33", "private_key":
        "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDBwKEjX8kbtqIO\nN
9\nPJ3rG1euV6aaNoKRcZF5wlRftyBF2+rC5xXZZyyAqk1rQ7x5Z6uiadtCT4oaDlDo\n4bRvX7GWY1nnEscJwoeXVJIiBx/L7fb1PUeZ
Fl3NkIQrm+q8fwsKxTN0FPiu5XWXcJBR\nBj8ErgzFjkIsV6uh4Vzfwar1QjKmsZ8PvSUBeLQ8AznX3XCDbXGbW7sYndQ59MXj\njiz/j
/RHw2+VWZJC2XJEEB+vnpF1dKVa31XXtwa9DZe9mCWlC3u48dvVJk8R7oaSfrCy\nKRlMRX6bEmtYfVatlpj1HQKnBnRO76E8XJVnKs6e
VsVO+iFW2TahKsCLFHPWuXK1ozn2\nQJgJCPlfcMesWYyItrBwQ0bGZdY3HdR8ZHOSf3kbV2WOqdrEJ9l+zkx9zD6kFmlU\n3rAptn1Rx
UHusfKAVUecOpL8ZSsFN16PZmNxExYVctsaW7INu24ZQSJ4ycQu3k7H/yWT\nsGvkJpkTTwhWvVlXJ0+0jwpyL8Gs2uvxG2iCu0ozOrtg
1uMYu3Zaks+OHBI2aYWJXrdZ\niSb6J5y5adWdop/GysCbcJusrNTf6pYNFqdXfSLGlk8yGrKbveMt3bvFxzKKeQCU\nm2zGI1XRg04jV
wKBgQDiMdeErVDHSE/nSH4wGvTuc4P4EaczW3c/i0QwMn2ihlNaemG0\n0iyXl2xPhuxNQ5GDOcn8WlR9IxqqIP37tdLKH65bgUH3hFnO
IahKFsNuLGYocwKBgCe/\nv5eJk1PR25rgSWBiycklK4rVeit5sK4K4zONy9m8gITvbkLsZta0EdJHcJvdEAsa\nH3EYUkFEa3z9fLsQ5
/k4hz50dONCJANKyfJ+CkKfCETtAoGAVYejfk8YINJWy9uybQT2\nJqK6knC3feBIaxXsNsWQnlTRxiOWZy1EXqEDINS2YHCcAkQZZ13e
UuwuXPAWttKLe2h3\nxTUf0rmVnuWOWIHpWYC90wg=\n-----END
        PRIVATE KEY-----\n", "client_email": "kubernetes-dns@witronix-sandbox.iam.gserviceaccount.com",
        "client_id": "103673866224213774648", "auth_uri": "https://accounts.google.com/o/oauth2/auth",
        "token_uri": "https://oauth2.googleapis.com/token", "auth_provider_x509_cert_url":
        "https://www.googleapis.com/oauth2/v1/certs", "client_x509_cert_url": "https://www.googleapis.com
gserviceaccount.com"
        }'
```

Helm Secrets

```
$ helm secrets view secrets.yaml
secretFiles:
    gcloud-credentials.json: '{ "type
        "private_key_id": "9cf4e25933
        "-----BEGIN PRIVATE KEY-----\
9\nPJ3rG1euV6aaNoKRcZF5wlRftyBF2+rC5x
Fl3NkIQrm+q8fwsKxTN0FPiu5XWXcJBR\nBj8
/RHw2+VWZJC2XJEEB+vnpF1dKVa31XXtwa9DZ
VsVO+iFW2TahKsCLFHPWuXK1ozn2\nQJgJCPI
UHusfKAVUecOpL8ZSsFN16PZmNxExYVctsaW7
1uMYu3Zaks+OHBI2aYWJXrdZ\niSb6J5y5adW
wKBgQDiMdeErVDHSE/nSH4wGvTuc4P4EaczW3
IahKFsNuLGYocwKBgCe/\nv5eJk1PR25rgSWB
/k4hz50dONCJANKyfJ+CkKfCETtAoGAVYejfk
UuwuXPAWttKLe2h3\nxTUf0rmVnuWOWIHpWYC
        PRIVATE KEY-----\n", "client_
        "client_id": "103673866224213
        "token_uri": "https://oauth2.
        "https://www.googleapis.com/c
gserviceaccount.com"
        }'
```

```yaml
{{- if .Values.secretFiles }}
apiVersion: v1
kind: Secret
metadata:
  name: {{ template "traefik.fullname" . }}-secrets
  labels:
    app: {{ template "traefik.name" . }}
    chart: {{ template "traefik.chart" . }}
    release: "{{ .Release.Name }}"
    heritage: "{{ .Release.Service }}"
type: Opaque
data:
{{- range $filename, $fileContents := .Values.secretFiles }}
  {{ $filename }}: {{ $fileContents | b64enc | quote }}
{{- end }}
{{- end }}
```

Helm Secrets

- Helmfile uses Helm Secrets Plugin which uses SOPS
- SOPS supports different types of Encryption
  - AWS KMS, GCP KMS, Azure Key Vault, PGP, etc
- Secrets.yaml file gets encrypted and saved in Git
- Use a Secret Resource and unencrypted Secrets.yaml to create safe, secure, reliable, repeatable and distributed way to create Resources

https://github.com/zendesk/helm-secrets

https://github.com/mozilla/sops

https://gist.github.com/twolfson/01d515258eef8bdbda4f

Order Dependencies

- Runs Helm apply times
- Run Helm deploy in specified order
- Deploy parts of Stack using labels and selectors
- Different ways to declare dependencies

Order Dependencies

Demo in Codefresh

- Deploy MyApp, Ingress, DNS, Namespace
- Using Secrets.yaml
- Started with cfstep-helmfile but needed to customize

https://codefresh.io/steps/step/helmfile

Templating Values Yaml Files

- DRYing out values files
- Use Templating to support ad-hoc environments
- Hierarchy, environment level values files, with default but ability to override.
- Use Env Vars in values.yaml file
- Run scripts to return values
- Expose all of Sprig / Go Templating
- Render out other files
- Use different Secrets.yaml for Different Environments

Demo in Codefresh

- Deploy Stack in 2 Environments
  - Dev
    - DNS in GCP / CloudDNS
    - External DNS Chart
  - Default / Ad-Hoc
    - Leverage Templating / Script to setup an .xip.io address

Environments

- Picture of Yaml with Multiple Environments
- Setup another environment, on the same K8s cluster
- Setup another environment in a different K8s cluster
- Leverage Templating for Ad-Hoc environments

Other Helmfile Features

- Repo Per Chart
- Modules
  - GitHub, S3, etc.
- Env Variables
  - Supports standard, default, required
- Top Level Helmfile to Deploy everything
- Use Helmfile with Kustomize

Mono Repo vs Repo per Chart

1. Talk about mono repo approach we've been taking, common b/c of public helms charts repo. Helmfile flexible, so you can setup how you want to
2. Discuss Repo per Chart and things like getting Helmfile from git instead of a local file.

Other Tools

- Look at Helm Plugins
  - Secrets, Diff
- Helmsman, Kustomize
- Check Ecosystem, lots of tools
- Other Tools / Charts
  - https://github.com/stakater/Reloader
  - https://github.com/kubernetes-sigs/external-dns
  - https://github.com/jetstack/cert-manager
  - https://github.com/vmware-tanzu/velero

Summary

- Simple Case Study
- Consider SDLC for Charts
- Tons of power with Sprig / Go Templating
- Try out Helmfile, or evaluate what's out there

Questions?

Schedule a no cost, 1:1 call with our cloud migration specialists at https://rootleveltech.com/

Signup for a FREE account with UNLIMITED builds

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