

Fabrizio Lazzaretti
Einführung in Helm:
Die bessere Art Kubernetes-
Anwendungen zu managen





Fabrizio Lazzaretti

- 6+ years of experience with Kubernetes
- 5+ years of experience with Helm
- 4+ years of experience with helmfile
- Cloud-, Integration- & Enterprise Architect

Cloud Native Computing Foundation (CNCF)

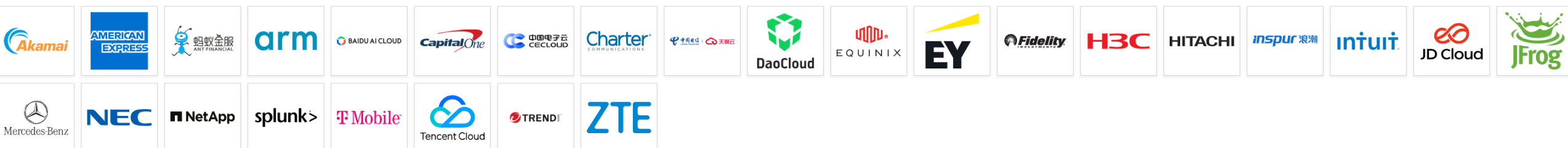
- CNCF is a foundation is a Linux Foundation project
- Enable open-source projects in the cloud & adds governance
- Founded in 2015 with the announcement of Kubernetes 1.0 (donation from Google, before called Borg)
- The big projects: Kubernetes, Prometheus, Envoy, ArgoCD

Sponsors:

Platinum (19)



Gold (27)



Helm: One of the Largest Growing Projects of CNCF (by Usage)

PROJECT USAGE TRENDS UPWARDS ACROSS INCUBATING AND GRADUATING PROJECTS

Largest Usage Gains (Growth In 2023) 2022 - 2023

35%



17%



kubernetes

24%



Prometheus

18%



etcd

25%



Helm is Used by 60% of the Represented Organizations in Production

Which Of These Graduated CNCF Projects Is Your Organization Using In Production Or Evaluating?



Kubernetes



- Open-source container orchestration system for automating software deployment
- Project is maintained by the Cloud Native Computing Foundation (CNCF)
- Uses the concept of “desired state”
- You normally use YAML files to deploy the resources
 - ⚡ Problem
 - Hard to keep track on what is important in huge files and what is different per use case
 - Low-level API
 - Copy-Paste Deployments (no parameters)
 - No simple rollback (delete is manual)

➔  solves these problems

What is HELM?

“A *helmsman* or “helm” is a person who steers a ship, sailboat, submarine, other types of maritime vessel, or spacecraft.” — [Wikipedia](#)



What is Helm? – apt get for Kubernetes

- Package manager for Kubernetes
- Graduated CNCF project
- A package is called “Chart”

Helm does `kubectl apply` & `kubectl delete` for you

- Create
- Update
- Delete

The Chart Structure

mychart/

```
Chart.yaml           # A YAML file containing information about the chart
LICENSE               # OPTIONAL: A plain text file containing the license
                      # for the chart
README.md             # OPTIONAL: A human-readable README file
requirements.yaml    # OPTIONAL: A YAML file listing dependencies for the chart
values.yaml         # The default configuration values for this chart
charts/               # A directory containing any charts upon which this chart
                      # depends.
templates/          # A directory of templates that, when combined with values,
                      # will generate valid Kubernetes manifest files.
templates/NOTES.txt  # OPTIONAL: A plain text file containing short usage notes
```

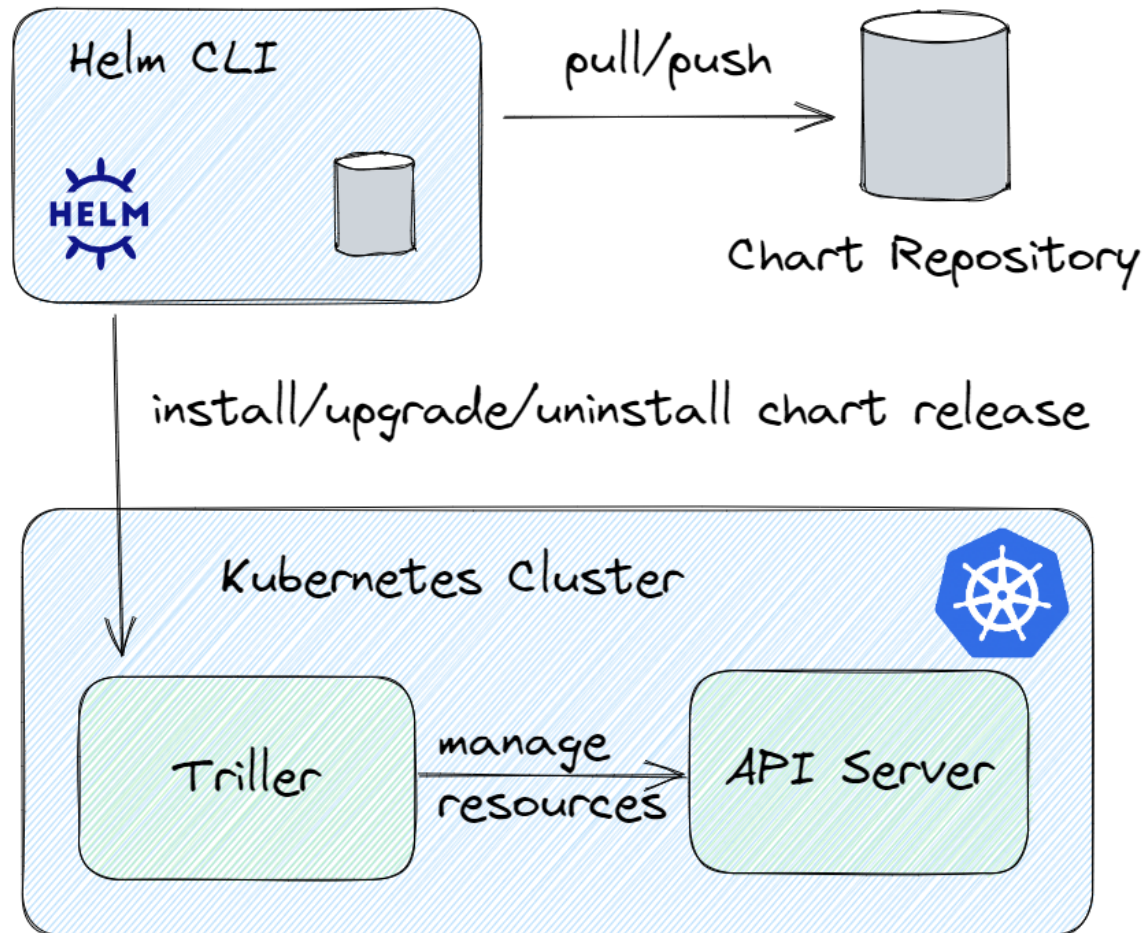
Templates

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ include "mychart.fullname" . }}-config-map
  labels:
    {{- include "mychart.labels" . | nindent 4 }}
data:
  config: {{ .Values.config }}
```

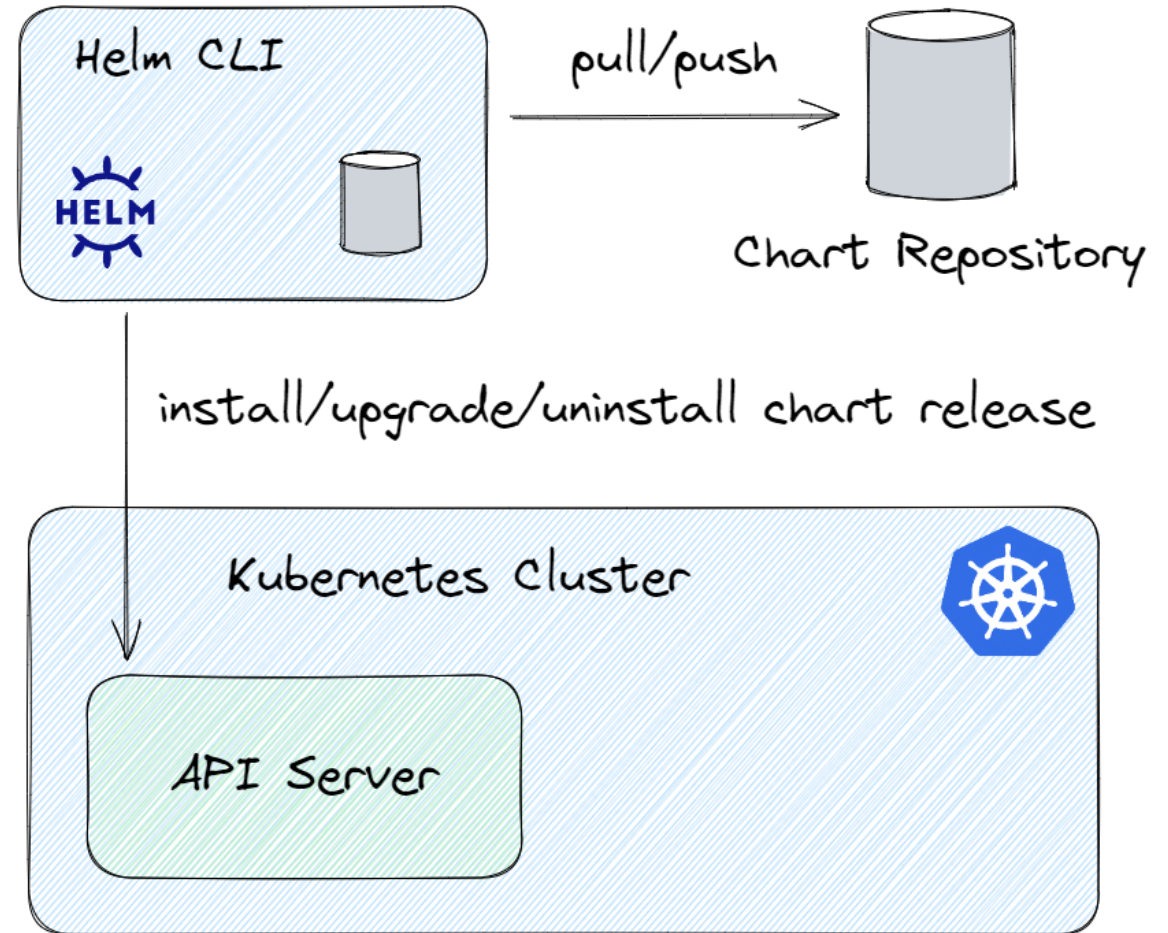
```
some:
  - configA
  - configB
```

Helm Moved with v3 to a Client-Only Implementation

Helm 2



Helm 3



Helm in Action:

Deploy Nginx

1. Find the Helm chart you want:

nginx: <https://artifacthub.io/packages/helm/bitnami/nginx>

2. Install the chart:

```
helm install hello-world  
oci://registry-1.docker.io  
/bitnamicharts/nginx
```

[Back to "nginx" results](#)

nginx

Helm chart

Networking

Bitnami

Bitnami

Star

79



NGINX Open Source is a web server that can be also used as a reverse proxy, load balancer, and HTTP cache. Recommended for high-demanding sites due to its ability to provide faster content.



SUBSCRIPTIONS: 23 WEBHOOKS: 3 PRODUCTION USERS: 2

nginx

Bitnami package for NGINX Open Source

NGINX Open Source is a web server that can be also used as a reverse proxy, load balancer, and HTTP cache. Recommended for high-demanding sites due to its ability to provide faster content.

[Overview of NGINX Open Source](#)

Trademarks: This software listing is packaged by Bitnami. The respective trademarks mentioned in the offering are owned by the respective companies, and use of them does not imply any affiliation or endorsement.

TL;DR

```
helm install my-release oci://registry-1.docker.io/bitnamicharts/nginx
```

Looking to use NGINX Open Source in production? Try [VMware Tanzu Application Catalog](#), the commercial edition of the Bitnami catalog.

Introduction

INSTALL

TEMPLATES

DEFAULT VALUES

VALUES SCHEMA

CHANGELOG

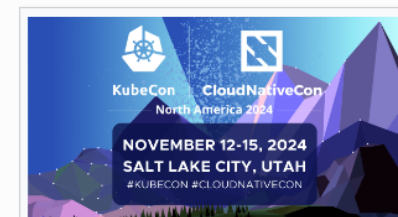
APPLICATION VERSION
1.27.2

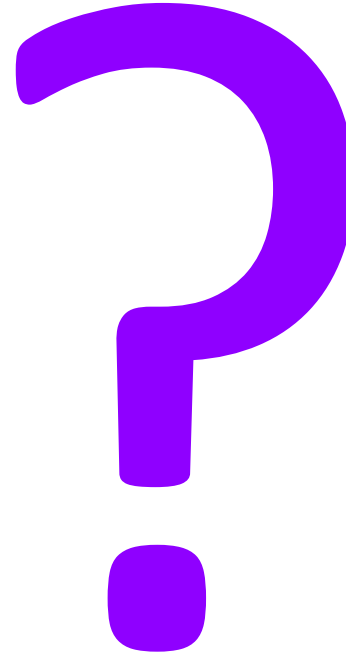
CHART VERSIONS

<https://artifacthub.io/>

```
helm install hello-world oci://registry-1.docker.io/bitnamicharts/nginx
```

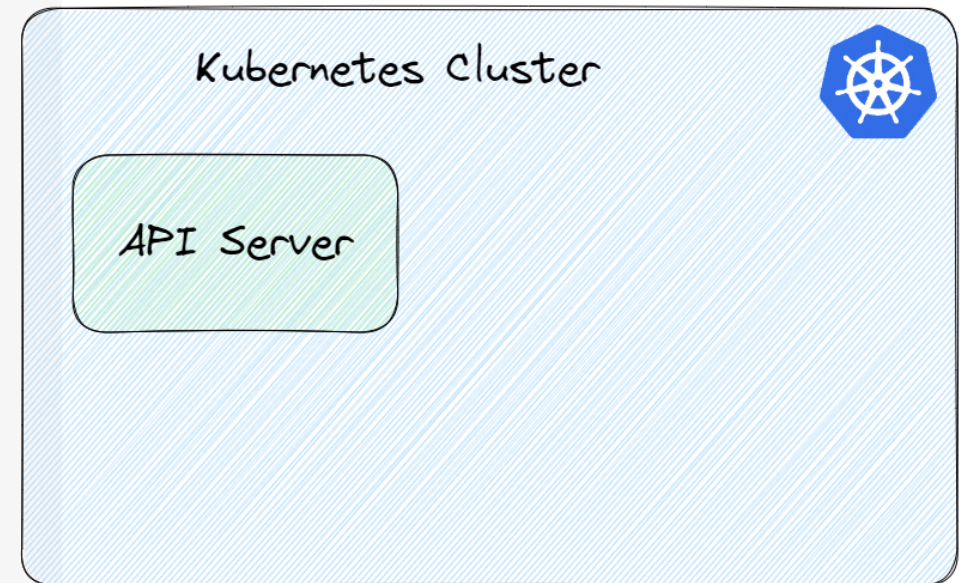
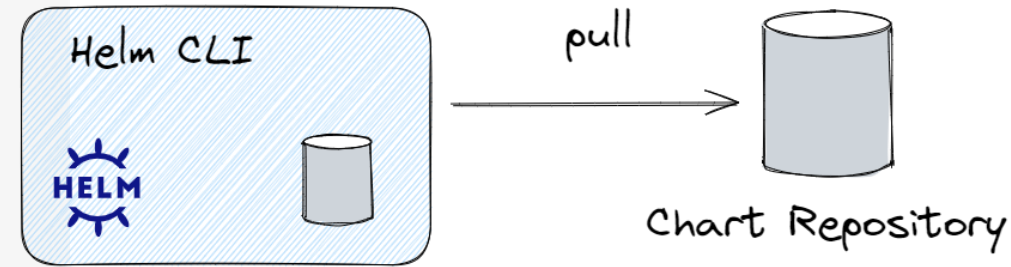
```
$ █
```

What was that?



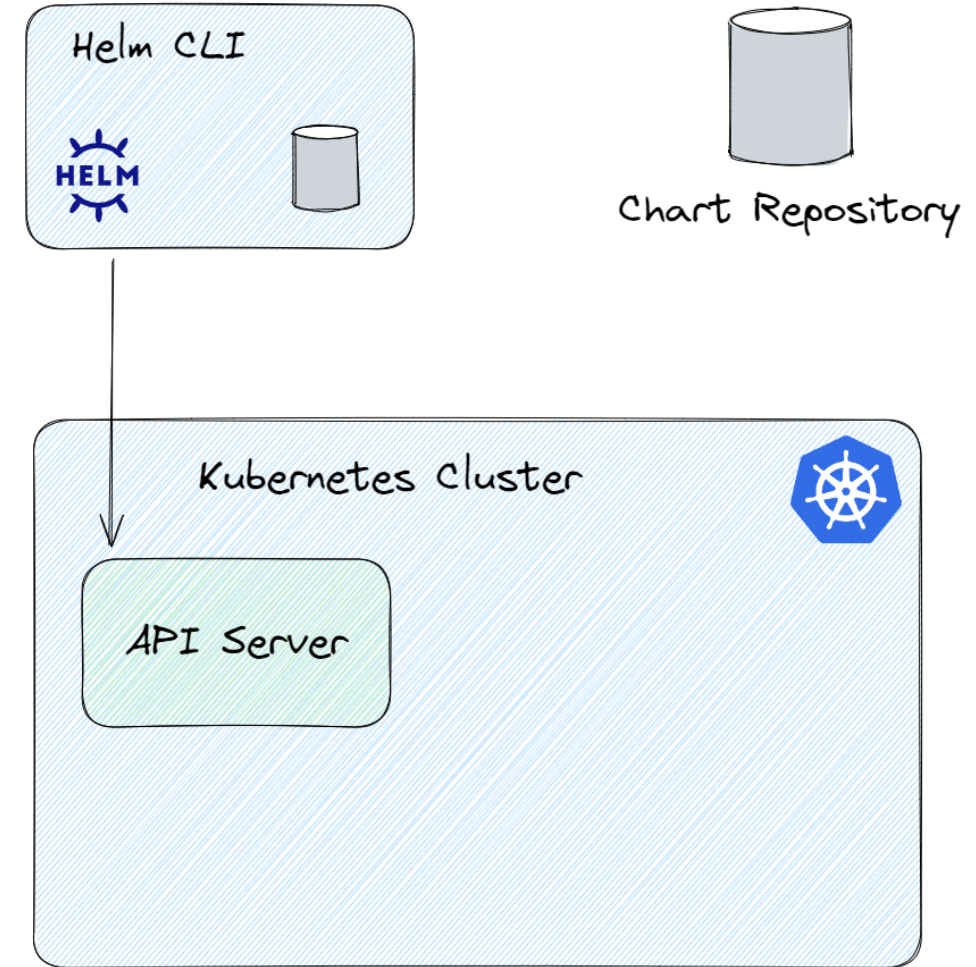
WAVESTONE

Deploying a Chart: Pull



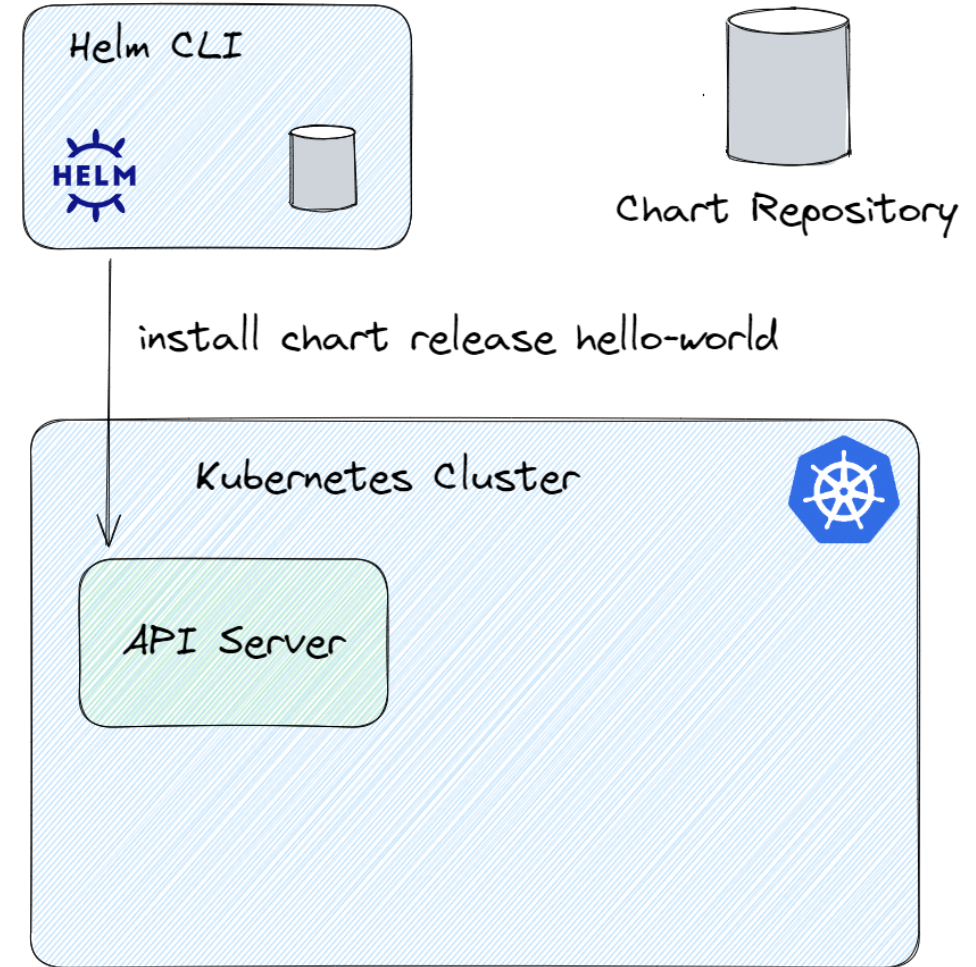
Deploying a Chart: Template

```
$ h
```



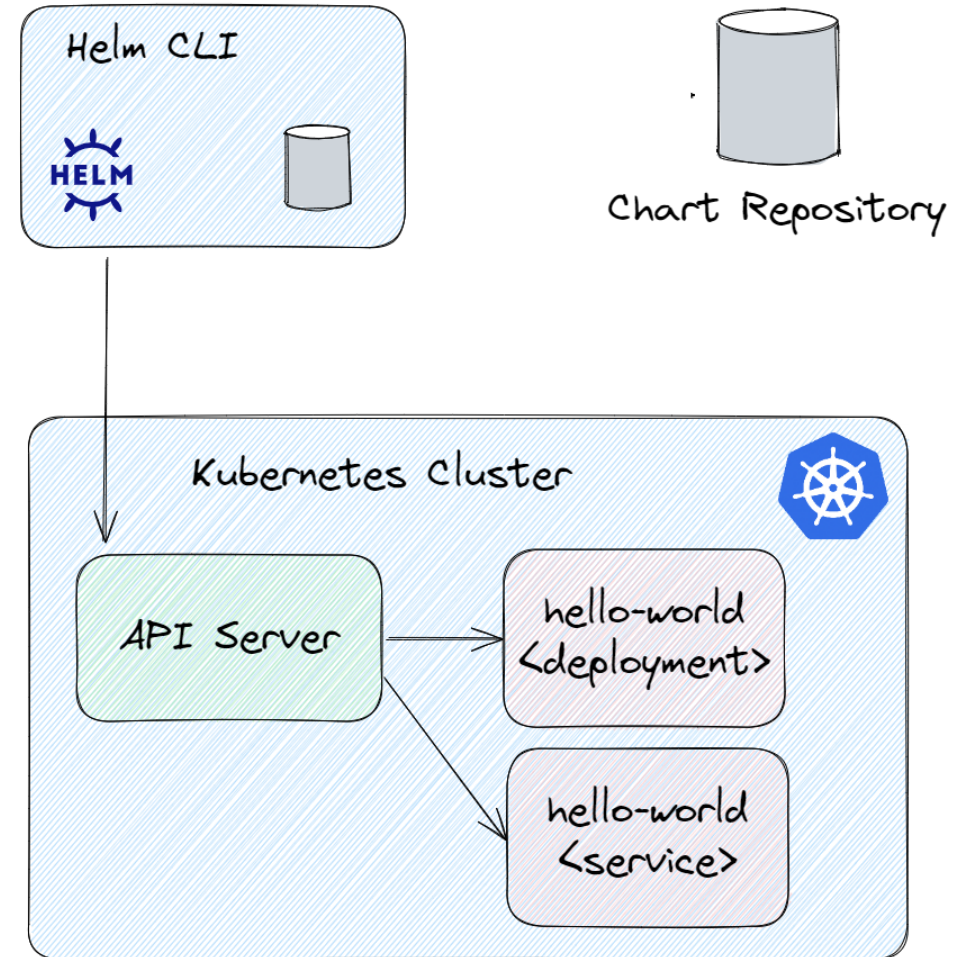
Deploying a Chart: Install

```
$ h
```



Deploying a Chart: The Deployment

```
$ kubectl get service
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)
hello-world-nginx   LoadBalancer       10.102.60.28    <pending>        80:32320/
kubernetes           ClusterIP           10.96.0.1       <none>           443/TCP
$ kubectl get deployments.app
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
hello-world-nginx   1/1      1             1            13m
$ kubectl get pod
NAME                READY    STATUS    RESTARTS    AGE
hello-world-nginx-8579d6c9c8-7574h  1/1     Running   0           13m
$ helm ls
NAME                NAMESPACE    REVISION    UPDATED
hello-world         default      1           2023-11-20 10:54:45.37137
```



Summary of the Steps

1. Download and unpack the Helm chart

```
helm pull oci://registry-1.docker.io/bitnamicharts/nginx --untar --untardir out/
```

2. See how it looks after it is rendered

```
helm template --output-dir './out' hello-world oci://registry-1.docker.io/bitnamicharts/nginx
```

3. Release

```
helm install hello-world oci://registry-1.docker.io/bitnamicharts/nginx
```

4. View the release

```
helm ls
```

Some of the Most Popular Charts on the Artifact Hub



kube-
prometheus-
stack



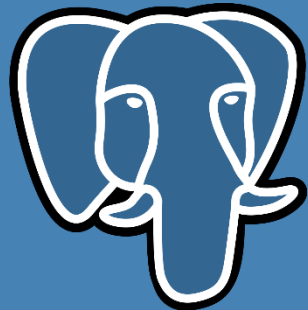
redis



cert-manager



harbor



postgresql

NGINX

nginx



argo-cd

Great!

... but we have requirements!

The image features two smooth, curved lines on a white background. A green curve starts at the left edge, dips slightly, and then rises towards the right. A purple curve starts from the bottom left, rises to cross the green curve, reaches a peak, and then slightly declines towards the right edge.

Customize the Chart with Parameters

- Add cli parameter

```
helm upgrade hello-world oci://registry-1.docker.io/bitnamicharts/nginx --set replicaCount=2
```

- Add a YAML file

```
helm upgrade hello-world oci://registry-1.docker.io/bitnamicharts/nginx -f hello-world-values.yaml
```

Helm Keeps the Previews Released Versions

1. Diff

```
helm diff upgrade hello-world oci://registry-1.docker.io/bitnamicharts/nginx -f hello-world-values.yaml
```

2. Upgrade

```
helm upgrade
```

3. Show versions & status

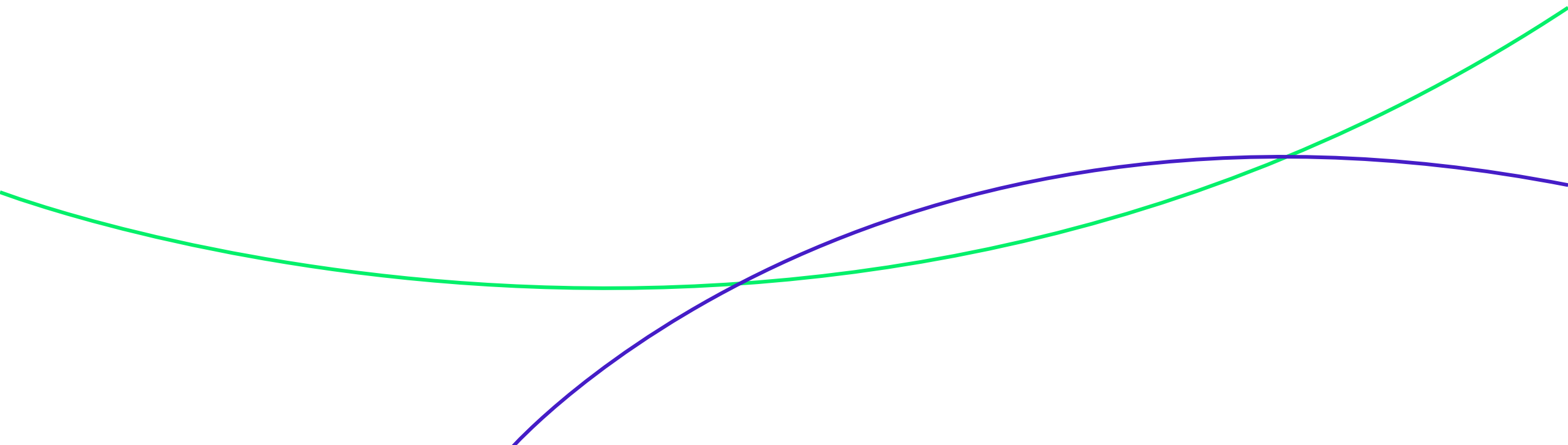
```
$ helm ls
```

NAME	NAMESPACE	REVISION	UPDATED	STATUS
hello-world	default	2	2023-11-20 12:18:44.022768466 +0000 UTC	deployed

4. Rollback

```
helm rollback hello-world
```

We need an own app!



Building an Own Chart

1. Create a chart

```
helm create mychart
```

2. Configure as needed

3. Deploy it

```
helm install myapp ./mychart
```

4. Publish (with OCI repository)

```
helm package ./mychart
```

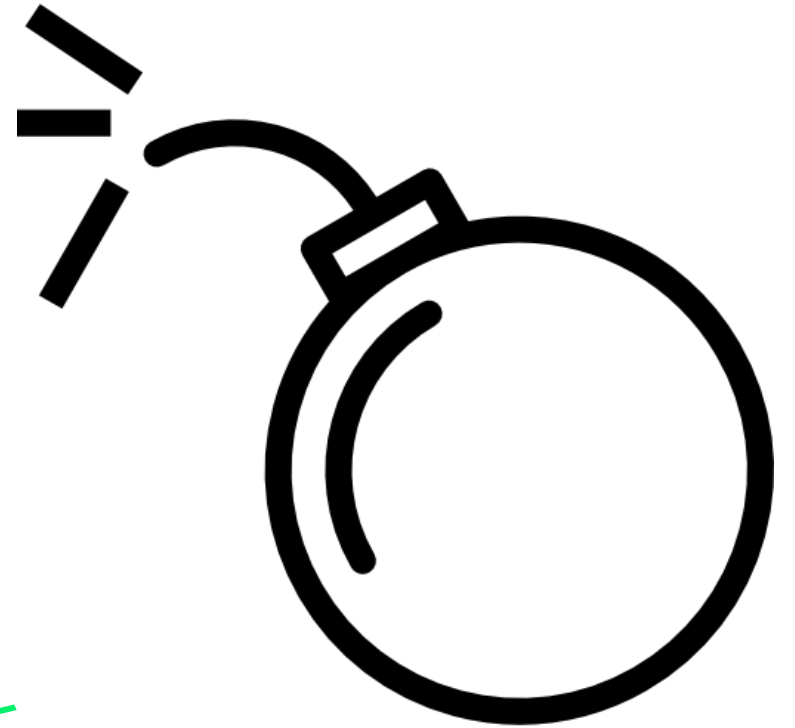
```
helm push mychart-0.1.0.tgz oci://my.registry/project/mychart
```

```
└─ mychart
  └─ charts
  └─ templates
    └─ _helpers.tpl
    └─ configmap.yaml
    └─ Chart.yaml
    └─ values.yaml
```

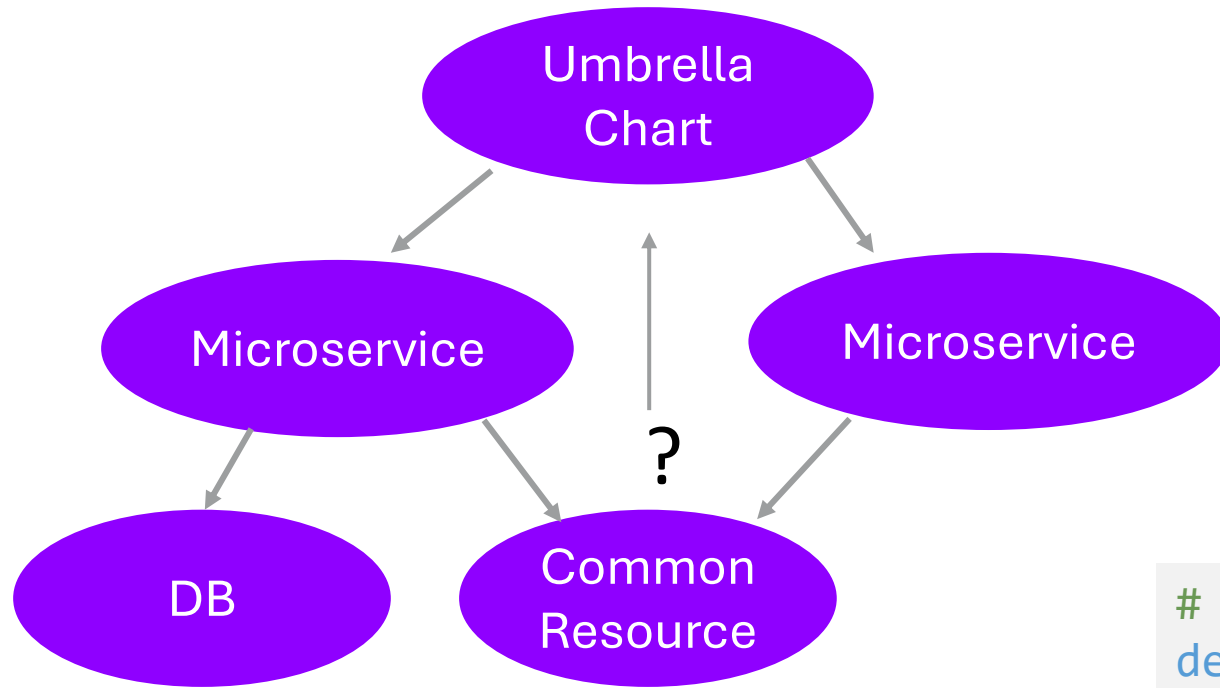
Many apps

many charts

Do we have a mess now?



Umbrella Charts



- 1 chart for everything
 - => 1 version, atomic chart

Problems:

- No isolation for names
- Minor inconvenience: install subcharts, ...

```
# umbrella-chart/Chart.yaml
dependencies:
- name: svc1
  repository: "https://charts.ex.com/repo"
  version: 1.2.3
- name: svc2
  repository: "https://charts.x.com/repo"
  version: 2.3.4
```

Multiple Environments

- `helm install mychart . -f values.yaml -f values-env.yaml`
- Helm will merge them right-to-left (right taking precedence).

```
mychart/  
values-dev.yaml  
values-prod.yaml  
values.yaml
```

Simplify with helmfile



Manage Helm with a Single helmfile

```
# helmfile.yaml
releases:
- name: hello-world
  namespace: default
  chart: oci://registry-1.docker...
  version: 15.4.2
  set:
  - name: replicaCount
    value: 2
```

```
$ helmfile apply
```

Manage Helm with a Single helmfile

- Keep separate Helm releases
 - => separate upgrade & rollback
 - With dependency awareness

```
# helmfile.yaml
releases:
- name: hello-world
  namespace: default
  chart: oci://registry-1.docker...
  version: 15.4.2
  set:
  - name: replicaCount
    value: 2
```

```
$ helmfile apply
```

Manage Helm with a Single helmfile

- Keep separate Helm releases
- Install dependencies into multiple namespaces

```
# helmfile.yaml
releases:
- name: hello-world
  namespace: default
  chart: oci://registry-1.docker...
  version: 15.4.2
  set:
  - name: replicaCount
    value: 2
```

```
$ helmfile apply
```

Manage Helm with a Single helmfile

- Keep separate Helm releases
- Install dependencies into multiple namespaces
- Environment support out of the box

```
environments:  
  production:  
    values:  
      - production.yaml  
  development:  
---  
releases:  
- name: hello-world  
  namespace: default  
  chart: oci://registry-  
1.docker.io/bitnamicharts/nginx  
  version: 15.4.2  
  set:  
  - name: replicaCount  
    value: 2
```

```
$ helmfile --environment production apply
```

Manage Helm with a Single helmfile

- Keep separate Helm releases
- Install dependencies into multiple namespaces
- Environment support out of the box
- Go templating for values and more
 - Make deployment simpler

```
releases:  
- name: chart-{{ .Values.myrelease.name }}  
  chart: mychart  
  version: {{ .Values.myrelease.version }}  
  values:  
  - values.yaml.gotmpl
```

```
$ helmfile apply
```

Manage Helm with a Single helmfile

- Keep separate Helm releases
- Install dependencies into multiple namespaces
- Environment support out of the box
- Go templating for values and more
- No difference between install/update
 - No helm repo add
 - No helm install & helm upgrade
 - => just helmfile apply

```
# helmfile.yaml
repositories:
  - name: prometheus-community
    url: https://prometheus-
community.gi...

releases:
  - name: prom-norbac-ubuntu
    namespace: prometheus
    chart: prometheus-community/prometheus
```

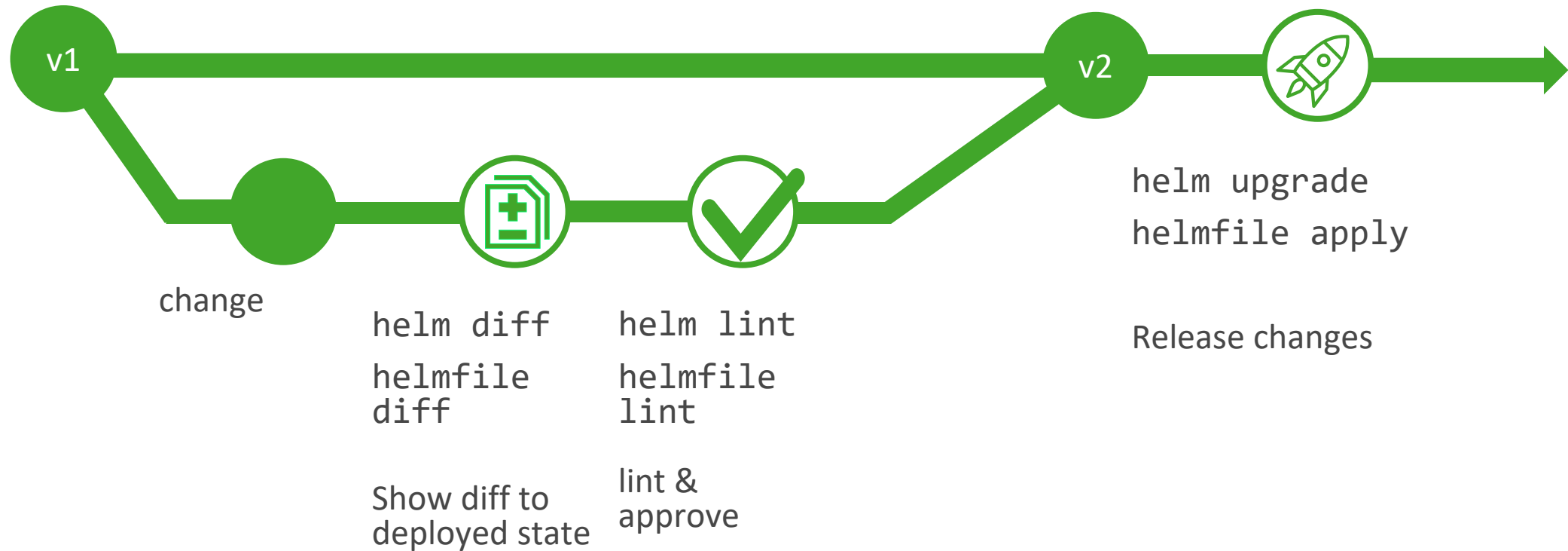
```
$ helmfile apply
```

CI/CD



GitOps Works Perfectly with Helm (& Helmfile)

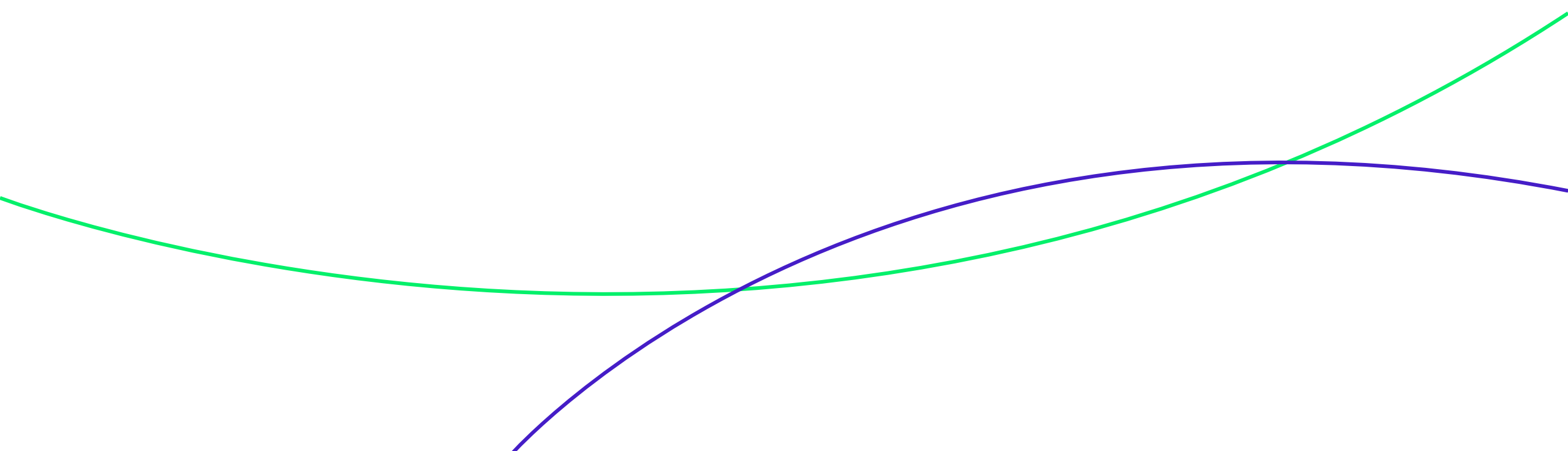
- Everything in code/config



Automate the Deployment with Terraform

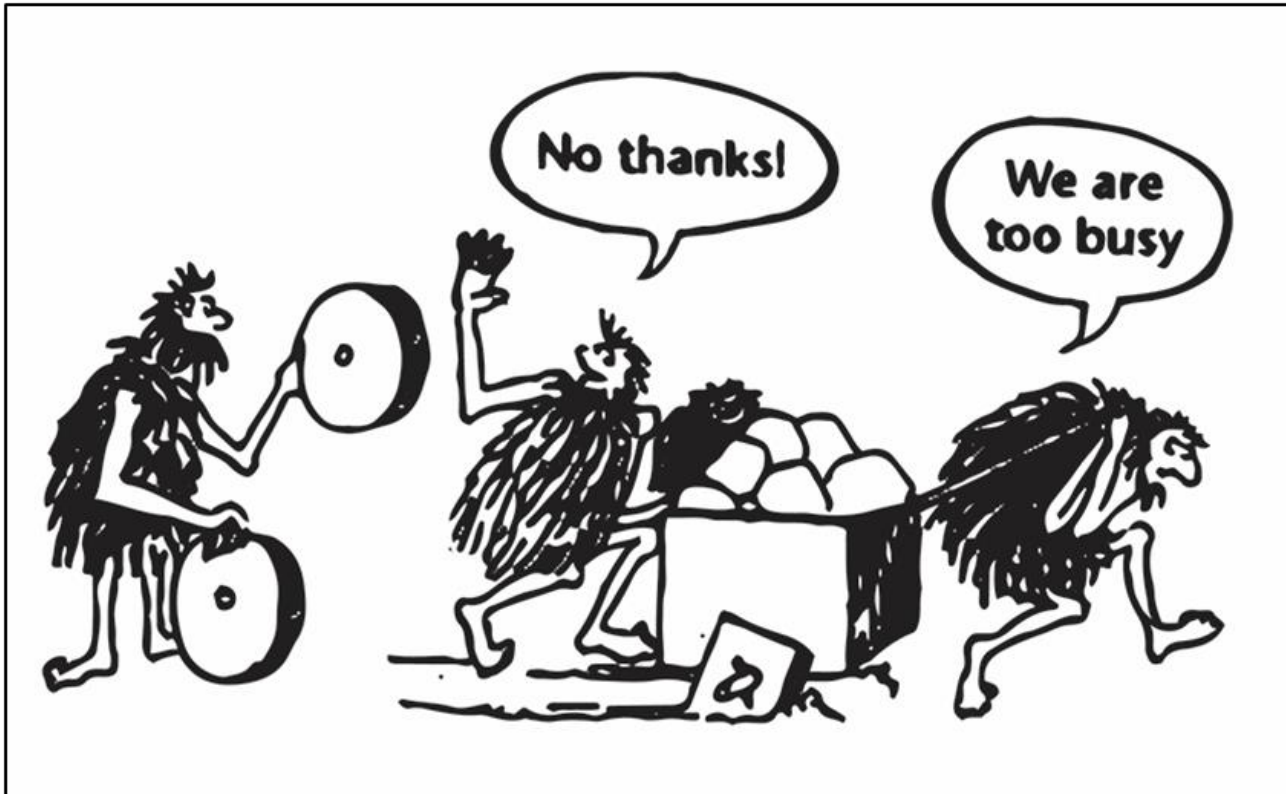
```
provider "helm" {  
  kubernetes {  
    config_path = "~/.kube/config"  
  }  
}  
  
resource "helm_release" "hello-world" {  
  name     = "hello-world"  
  chart    = "oci://registry-1.docker.io/bitnamicharts/nginx"  
  
  set {  
    name = "replicaCount"  
    value = "2"  
  }  
}
```

Conclusion



Helm & Helmfile simplify your life

1. Helm simplifies your life
2. Helmfile simplifies your life even more ;)



<https://github.com/Lazzaretti/helm-demo>

#WeAreWavestone

Your most trusted partner for strategic transformations.