

A large wooden ship's wheel is mounted on a wall with a textured, peeling white surface. Several framed pictures are hanging on the wall around the wheel. The wheel has a dark wooden rim and lighter wooden spokes. A brass-colored hub is in the center. The scene is dimly lit, with a warm glow from a light source on the right.

Kubernetes與Helm的 愛恨情仇

Hello! I am Billy Yang

k8s, Golang and Unsupervised learning

Trend Micro
Backend Engineer

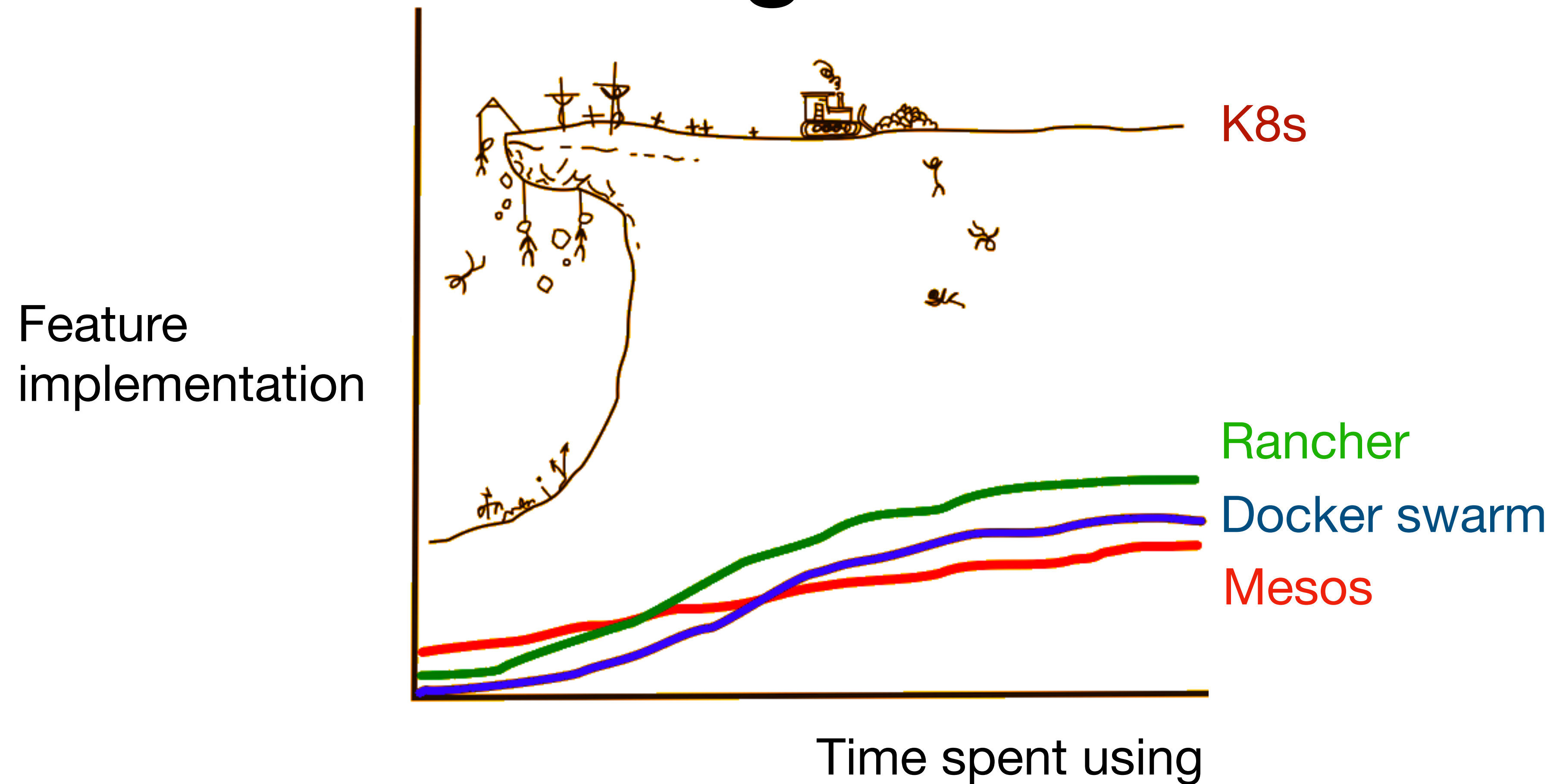


Agenda

- Why we need Helm?
- What is Helm?
- How do we use Helm?
- What's news in Helm?



Container Orchestration Learning Curves



Why K8s Is Hard

- Many small and interconnected components (k8s resource)
- configuration is relatively complex
- Documentation isn't really good

60 resource types in k8s v1.12.5

NAME	SHORTNAMES	APIGROUP	NAMESPACED	KIND
bindings			true	Binding
componentstatuses	cs		false	ComponentStatus
configmaps	cm		true	ConfigMap
endpoints	ep		true	Endpoints
events	ev		true	Event
limitranges	limits		true	LimitRange
namespaces	ns		false	Namespace
nodes	no		false	Node
persistentvolumeclaims	pvc		true	PersistentVolumeClaim
persistentvolumes	pv		false	PersistentVolume
Pods	po		true	Pod
podtemplates			true	PodTemplate
replicationcontrollers	rc		true	ReplicationController
resourcequotas	quota		true	ResourceQuota
secrets			true	Secret
serviceaccounts	sa		true	ServiceAccount
services	svc		true	Service
mutatingwebhookconfigurations		admissionregistration.k8s.io	false	MutatingWebhookConfiguration
validatingwebhookconfigurations		admissionregistration.k8s.io	false	ValidatingWebhookConfiguration
customresourcedefinitions	crd,crds	apiextensions.k8s.io	false	CustomResourceDefinition
apiservices		apiregistration.k8s.io	false	APIService
controllerrevisions		apps	true	ControllerRevision
daemonsets	ds	apps	true	DaemonSet
deployments	deploy	apps	true	Deployment
replicasets	rs	apps	true	ReplicaSet
statefulsets	sts	apps	true	StatefulSet
tokenreviews		authentication.k8s.io	false	TokenReview
localsubjectaccessreviews		authorization.k8s.io	true	LocalSubjectAccessReview
selfsubjectaccessreviews		authorization.k8s.io	false	SelfSubjectAccessReview
selfsubjectrulesreviews		authorization.k8s.io	false	SelfSubjectRulesReview
subjectaccessreviews		authorization.k8s.io	false	SubjectAccessReview
horizontalpodautoscalers	hpa	autoscaling	true	HorizontalPodAutoscaler
cronjobs	cj	batch	true	CronJob
jobs		batch	true	Job
certificatesigningrequests	csr	certificates.k8s.io	false	CertificateSigningRequest
leases		coordination.k8s.io	true	Lease
events	ev	events.k8s.io	true	Event
daemonsets	ds	extensions	true	DaemonSet
deployments	deploy	extensions	true	Deployment
ingresses	ing	extensions	true	Ingress

We need a tool to
simplify k8s configuration writing and
make deployment easier

Why We ❤️ Helm ?

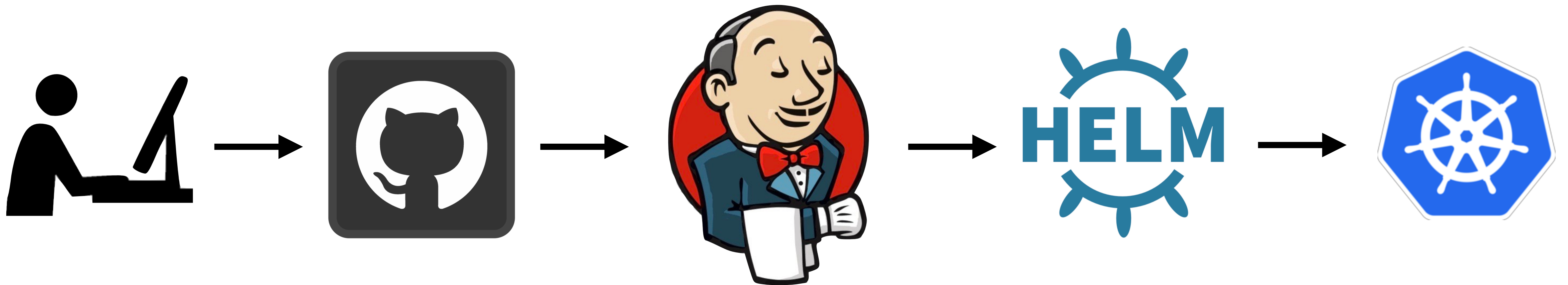
- Templating engine
- Rollback a service quickly and version support
- Installation and setup are simple
- Community and document are not bad

```
int, atom2-cms, Deployment (apps) has changed:
# Source: atom2-cms/templates/deployment.yaml
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: atom2-cms
  namespace: int
  labels:
    app: atom2-cms
    chart: atom2-cms
    release: int-atom2-cms
    heritage: Tiller
spec:
-   replicas: 1
+   replicas: 2
```

Objectives

- **Easy** to deploy it, just git push
- **Easy** to backup the deployed configuration file
- **Easy** to maintain or troubleshoot it
- **Not easy** to hack our k8s, make it as secure as possible

Blueprint



- Why we need Helm?
- **What is Helm?**
- How do we use Helm?
- What's news in Helm?



Helm

The Package Manager For K8s

Let's Write Helm Package

Chart : Helm Package

- A description of the package, Chart.yaml
- One or more template, which contains k8s resources
- After install a chart, it will create a **release**.
- Upgrade the release, it will increase **revision number**.



Chart.yaml

- Major information of Helm package

```
apiVersion: The chart API version, always "v1" (required)
name: The name of the chart (required)
version: A Semantic-versioning v2 version (required)
description: A single-sentence description of this project (optional)
```

- Semantic versioning



- `helm upgrade/push --force`

Go Template + YAML

template/deployment.yaml

```
apiVersion: apps/v1beta2
kind: Deployment
spec:
  replicas: {{ .Values.replicaCount }}
  template:
    spec:
      containers:
        - name: {{ .Values.app.name }}
          image:
            "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
            imagePullPolicy: IfNotPresent
            command: [ "/go/bin/hello-app" ]
```

values.yaml

```
replicaCount: 2
app:
  name: hello-app
image:
  repository:
    registry.us-west-1/app/
    hello-app
  tag: 1.0.0
```



**When DevOps
Engineers Use Helm Template**

Required & Default

- Basic

```
{{ required "log level is required" .Values.logLevel }}
```

```
{{ default "INFO" .Values.logLevel }}
```

- Default Check for Flat Values

```
logLevel: DEBUG  
logRotate: 30
```

```
logger:  
  level: {{ default "INFO" .Values.logLevel }}  
  rotate: {{ default 7 .Values.logRotate }}
```

Required & Default

- Default Check for Nested Values

```
logger:  
  level: DEBUG  
  rotate: 30
```

```
{{- if .Values.logger -}}  
logger:  
  level: {{ default "INFO" .Values.logger.level }}  
  rotate: {{ default 7 .Values.logger.rotate }}  
{{- else -}}  
logger:  
  level: ERROR  
  rotate: 1  
{{- end -}}
```

Make Type Clear

- Large integers will get converted to scientific notation in some cases.

```
packageId: 12345678
```

```
{{.Values.packageId }}
```

```
1.2345678e+07
```

```
packageId: "12345678"
```

```
{{ int .Values.packageId }}
```

```
12345678
```

```
packageId: "12345678912345678912"
```

```
{{ int .Values.packageId }}
```

```
0
```

Quote Value Or Template ?

```
logLevel: ""  
logRotate: 7
```

```
logLevel: {{ .Values.logLevel }}  
logRotate: {{ .Values.logRotate }}
```

```
logLevel:  
logRotate: 7
```

```
logLevel: "\nlogRotate: 12345678"  
logRotate: 7
```

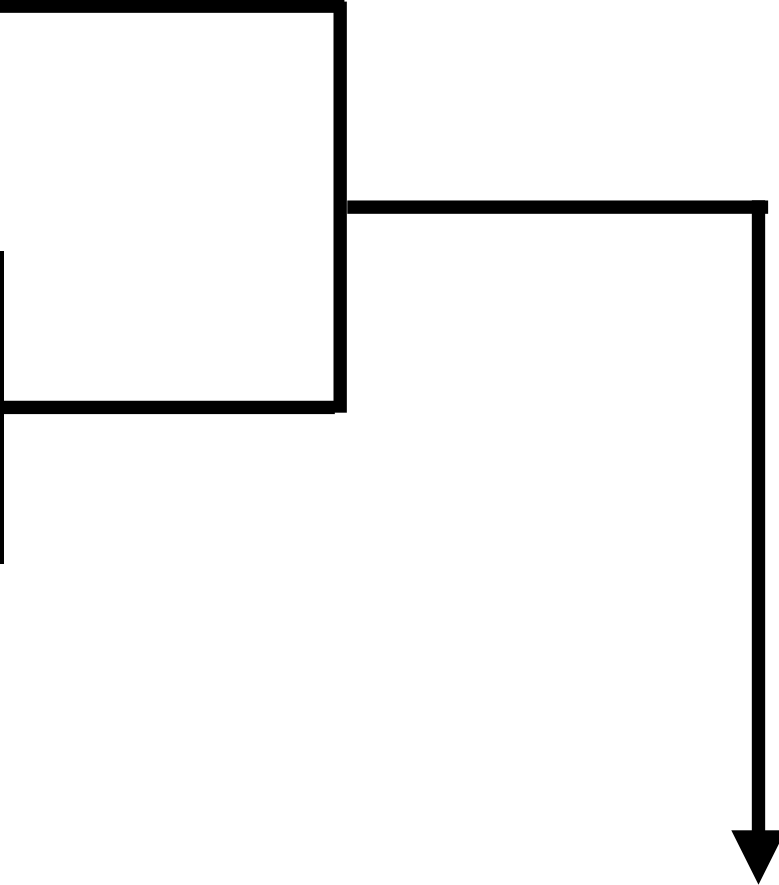
```
logLevel: {{ .Values.logLevel }}  
logRotate: {{ .Values.logRotate }}
```

```
logLevel:  
logRotate: 1.2345678e+07  
logRotate: 7
```

Quote Value Or Template ?

```
logLevel: '\nlogRotate: 12345678'  
logRotate: 7
```

```
logLevel: {{ .Values.logLevel | quote }}  
logRotate: {{ .Values.logRotate }}
```

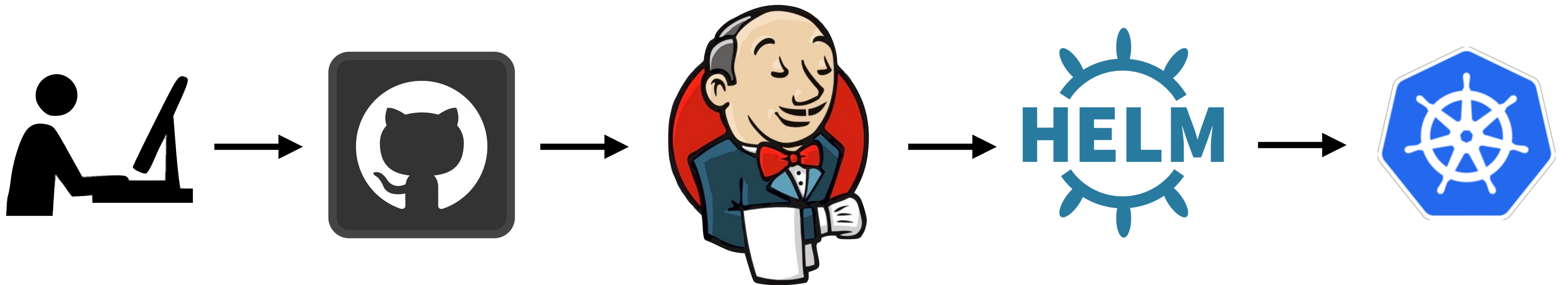


```
logLevel: "\nlogRotate: 1.2345678e+07"  
logRotate: 7
```

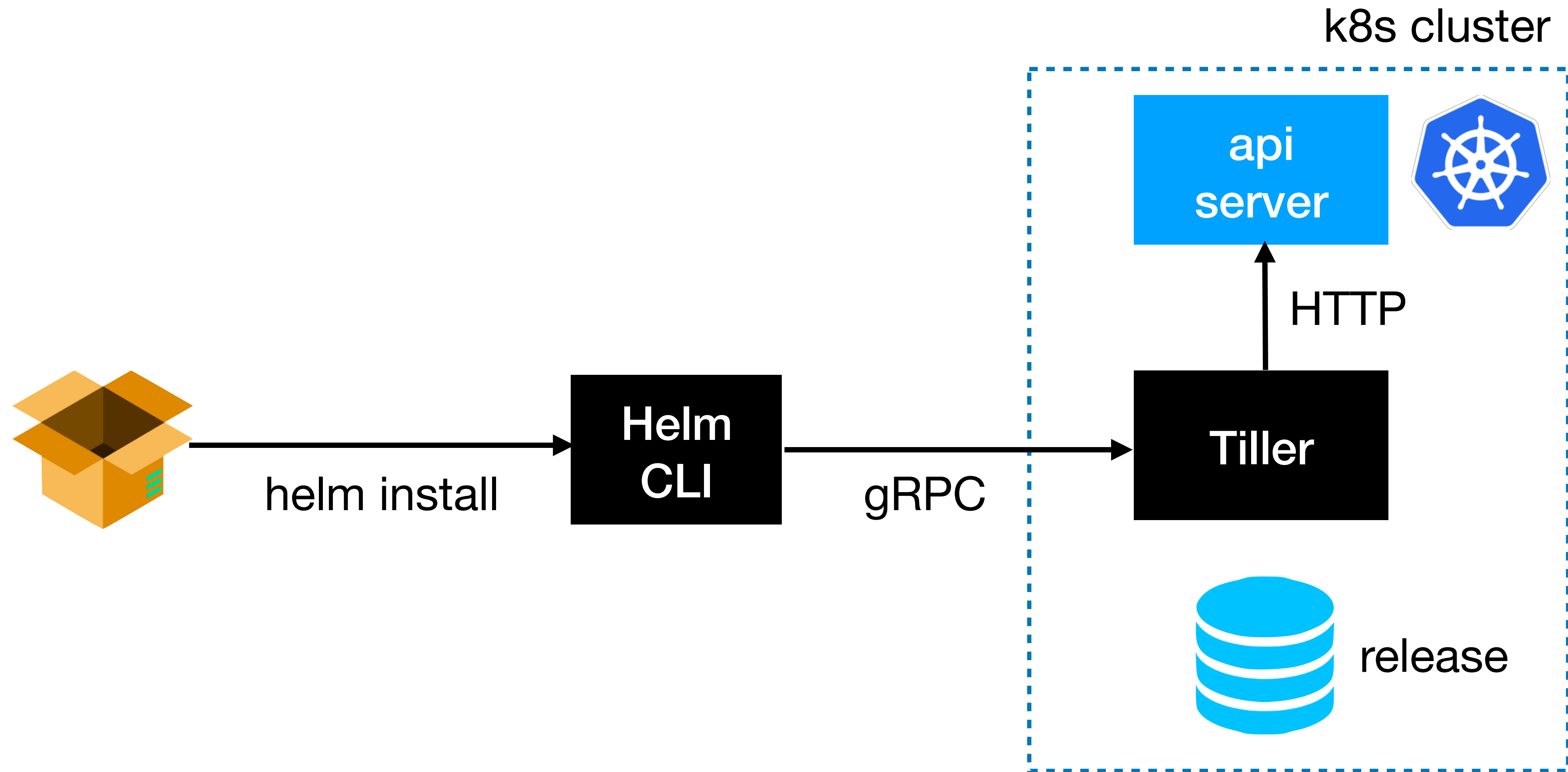
- Why we need Helm?
- What is Helm?
- How do we use Helm?
- What's news in Helm?



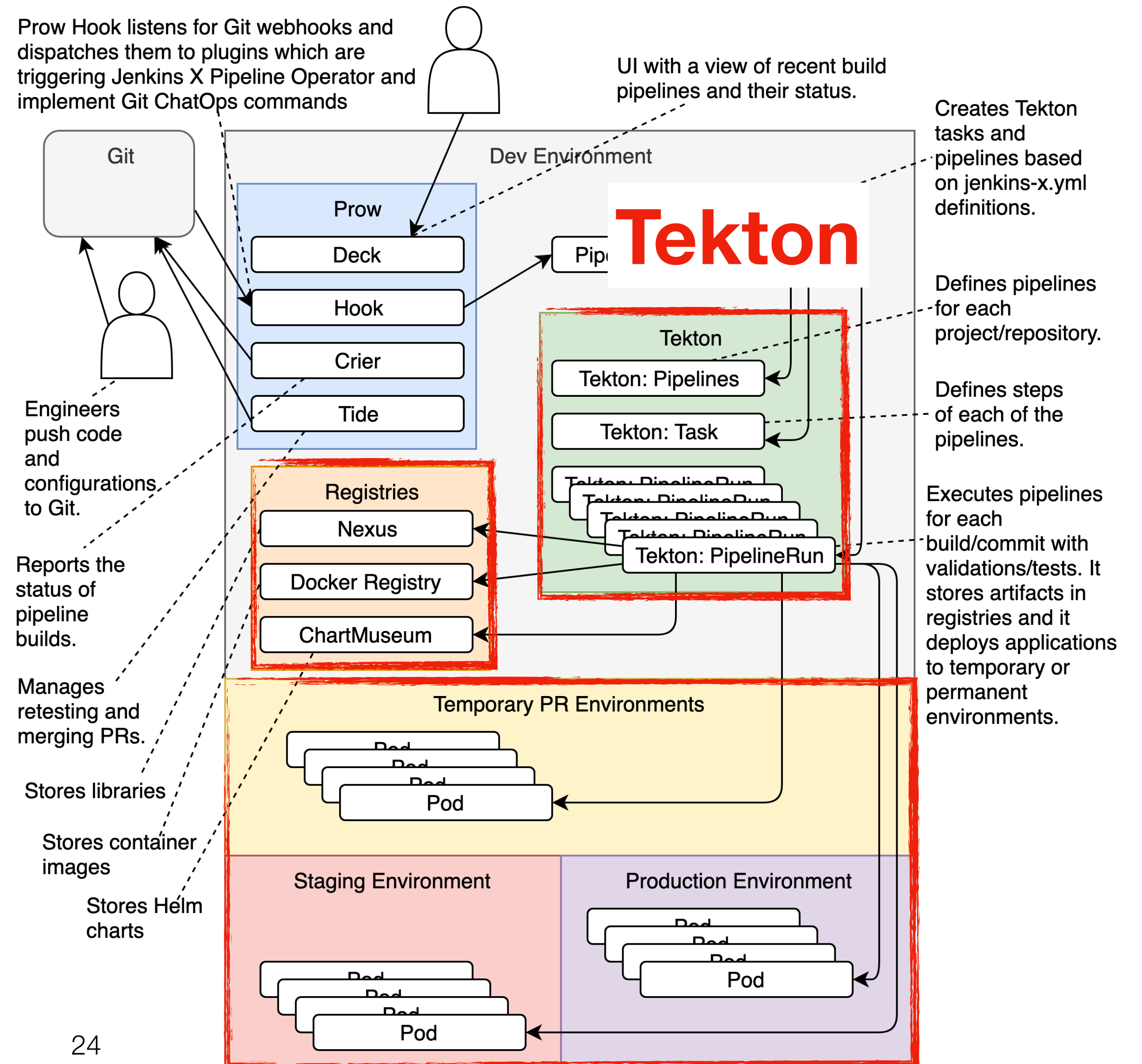
Blueprint



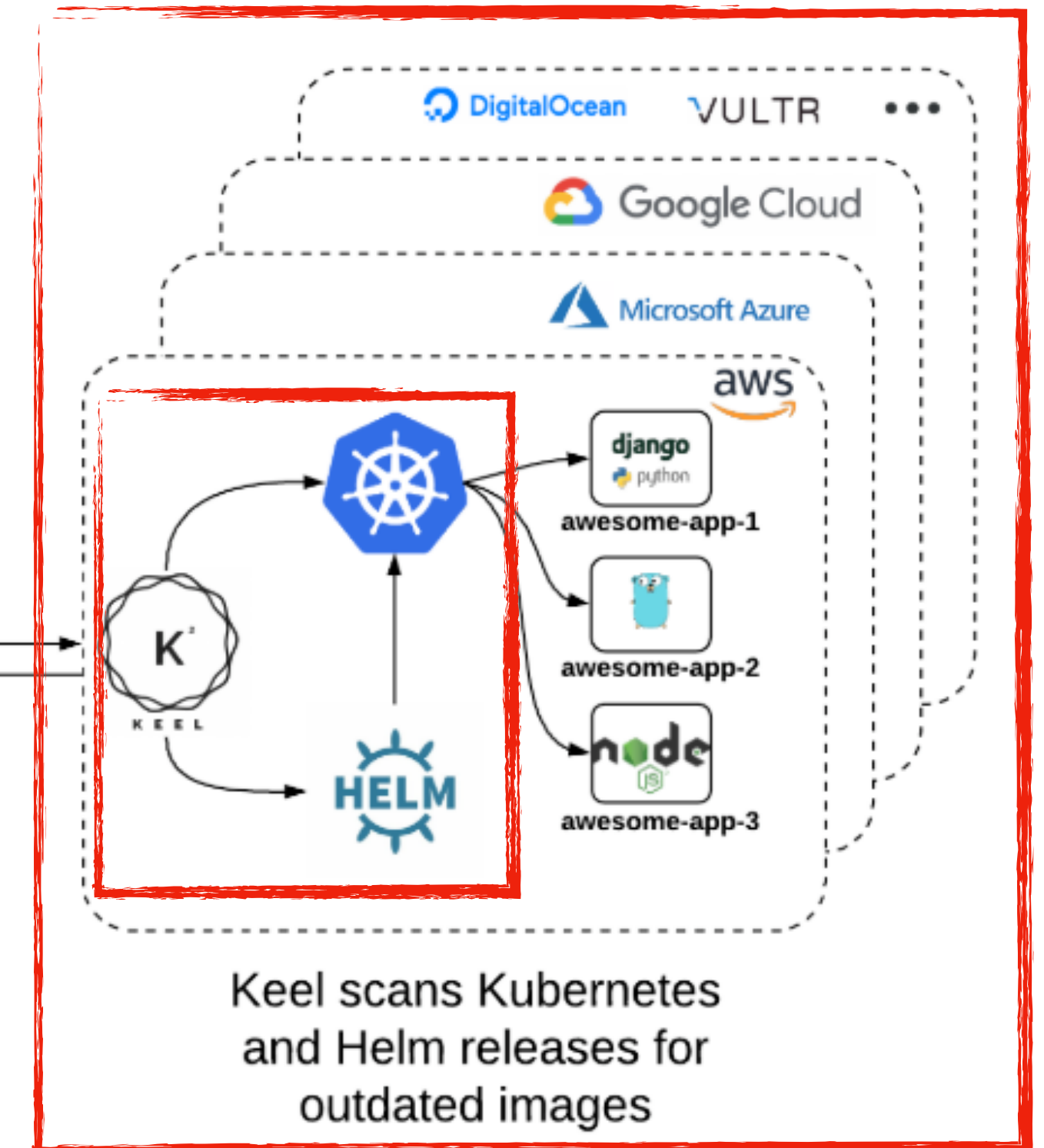
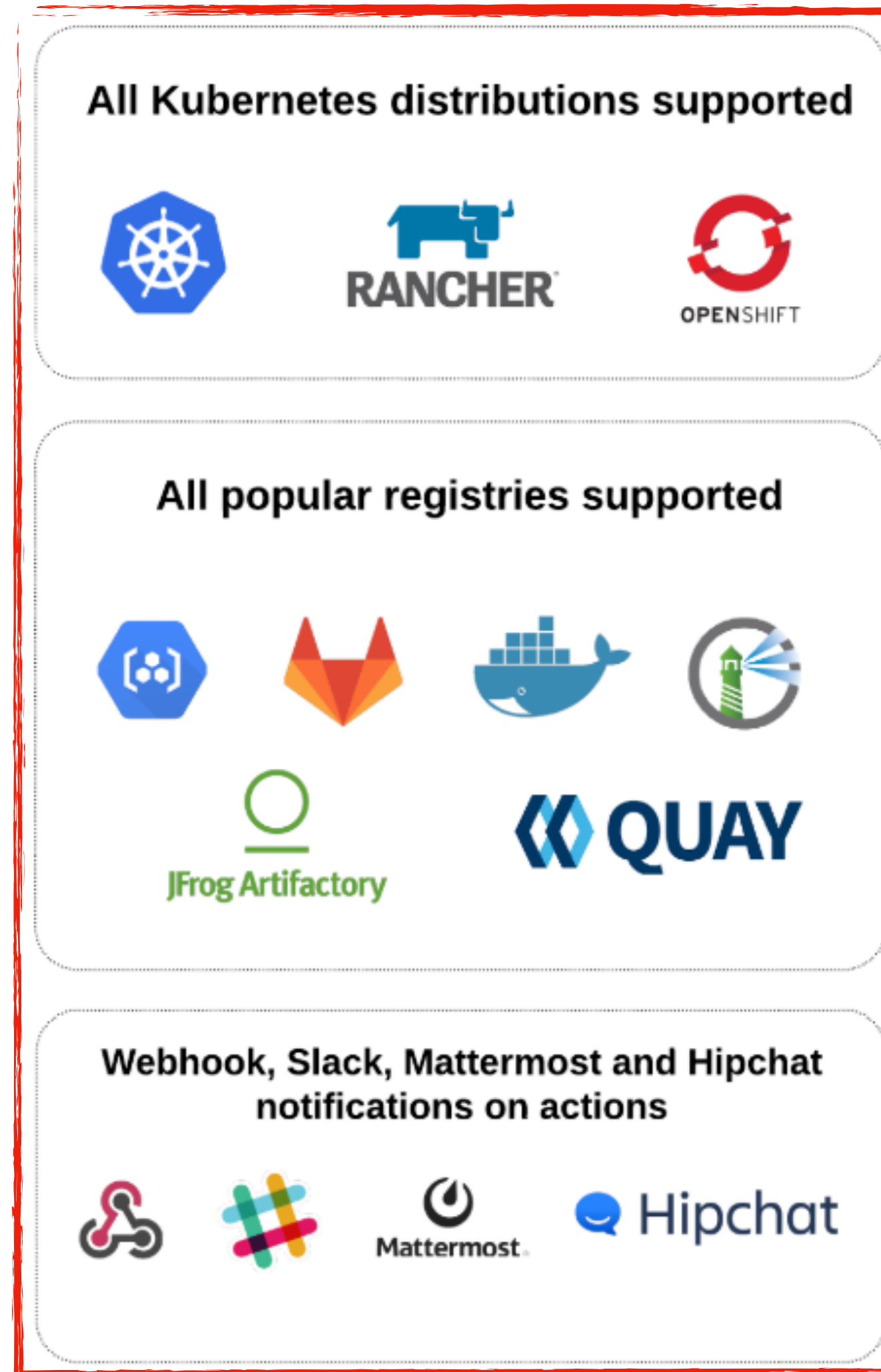
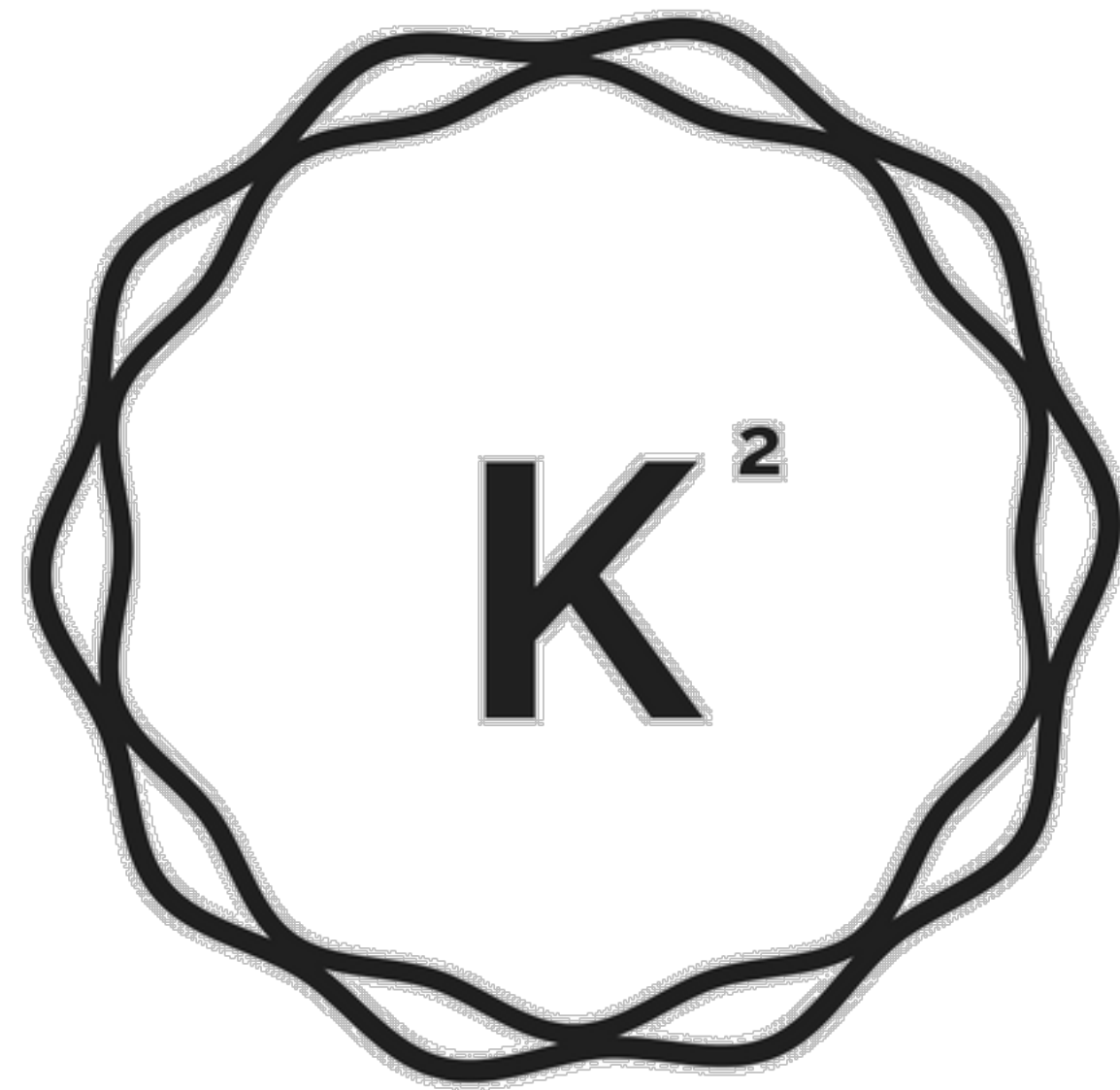
Helm Architecture



Jenkins X



Keel



**Build our Helm deployment pipeline
in Jenkins
without additional dependency**

Tasks

Update
Chart Version,
Image Tag, ...



Verify
Template
Rendering



Backup
Released
Chart



Deploy



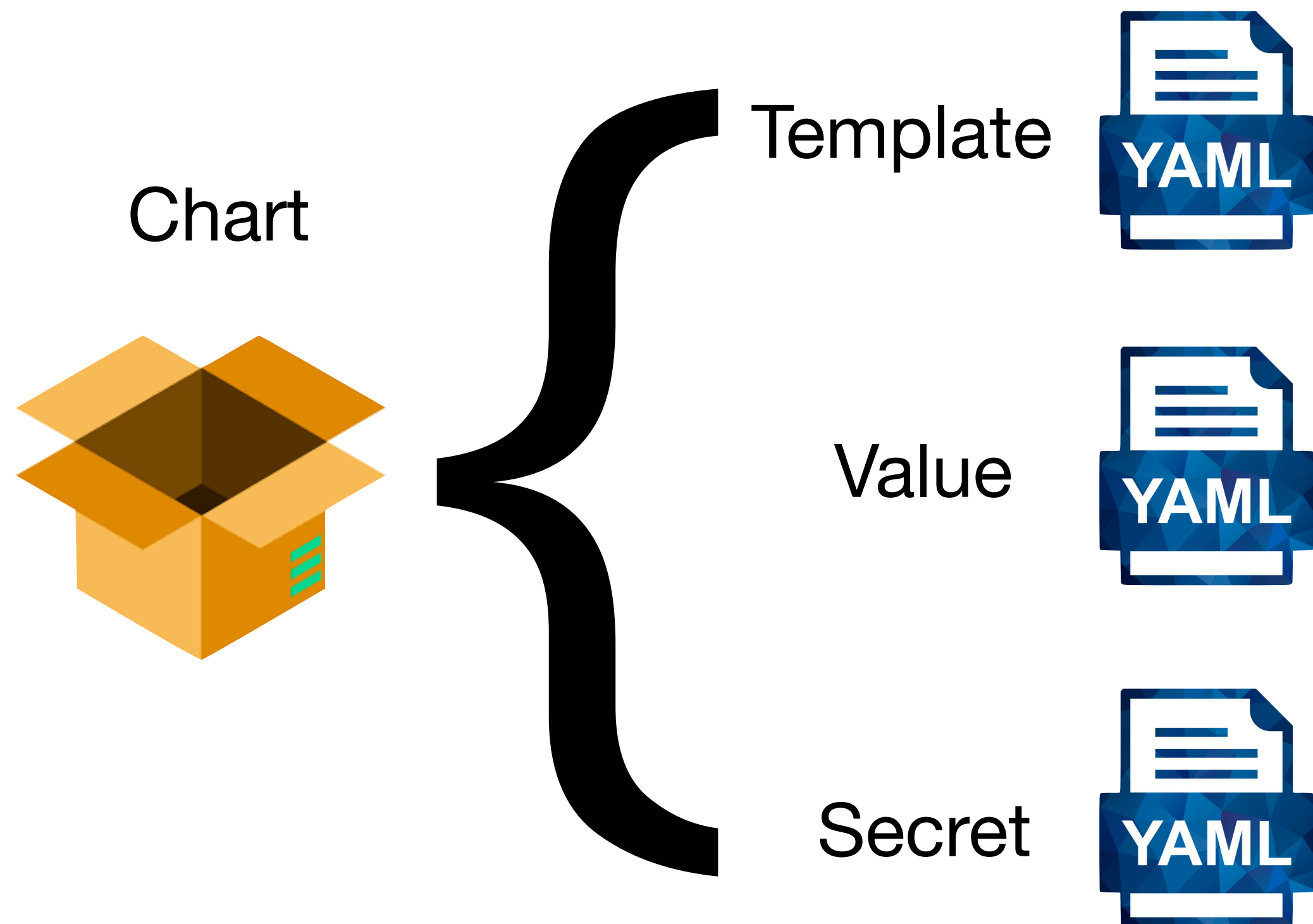
Check
Service
Status



Encrypt
Secret
in Git



Secret Management In Git



```
appValue:  
  db:  
    name: helldb  
    connectionTimeout: 60
```

```
appSecret:  
  db:  
    username: dbuser  
    password: "db@PASSWORD123"
```

Chart Encryption Tool

- We use **Helm-secret** plugin.
- Helm-secret uses **mozilla/sops (Secrets OPerationS)**
- Sops is an editor of encrypted files
 - Format support : YAML, JSON, INI, ...
 - Key provider support : AWS KMS, GCP KMS, Azure Key Vault and PGP

Chart Encryption Tool

```
appSecret:  
  db:  
    username: dbuser  
    password: "db@PASSWORD123"
```

```
sops -e -i  
secret.alpha.yaml
```

```
appSecret:  
  db:  
    username:  
ENC[AES256_GCM, data:CwE401s=, iv:2k=, aad:o=, tag:w==]  
    password:  
ENC[AES256_GCM, data:p673w==, iv:YY=, aad:UQ=, tag:A=]
```

Chart Encryption Flow

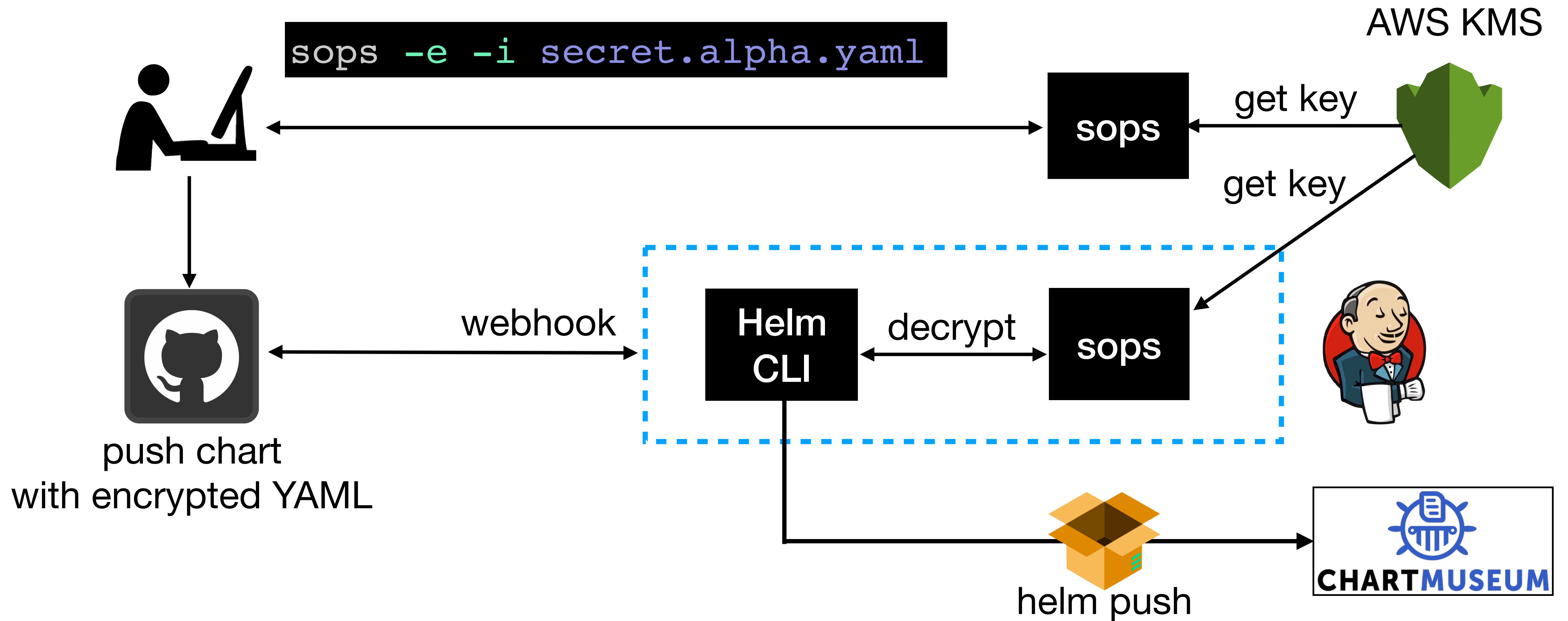
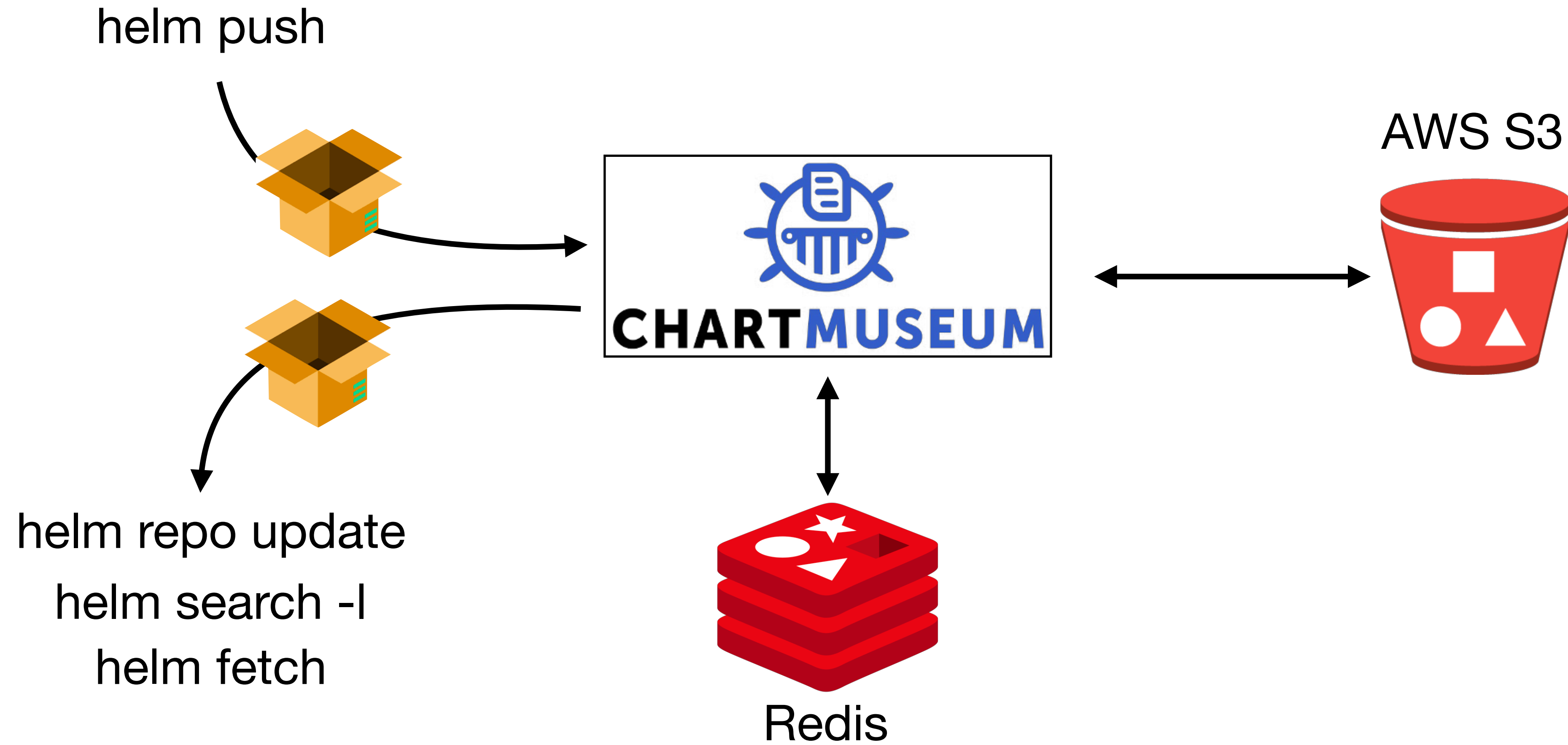
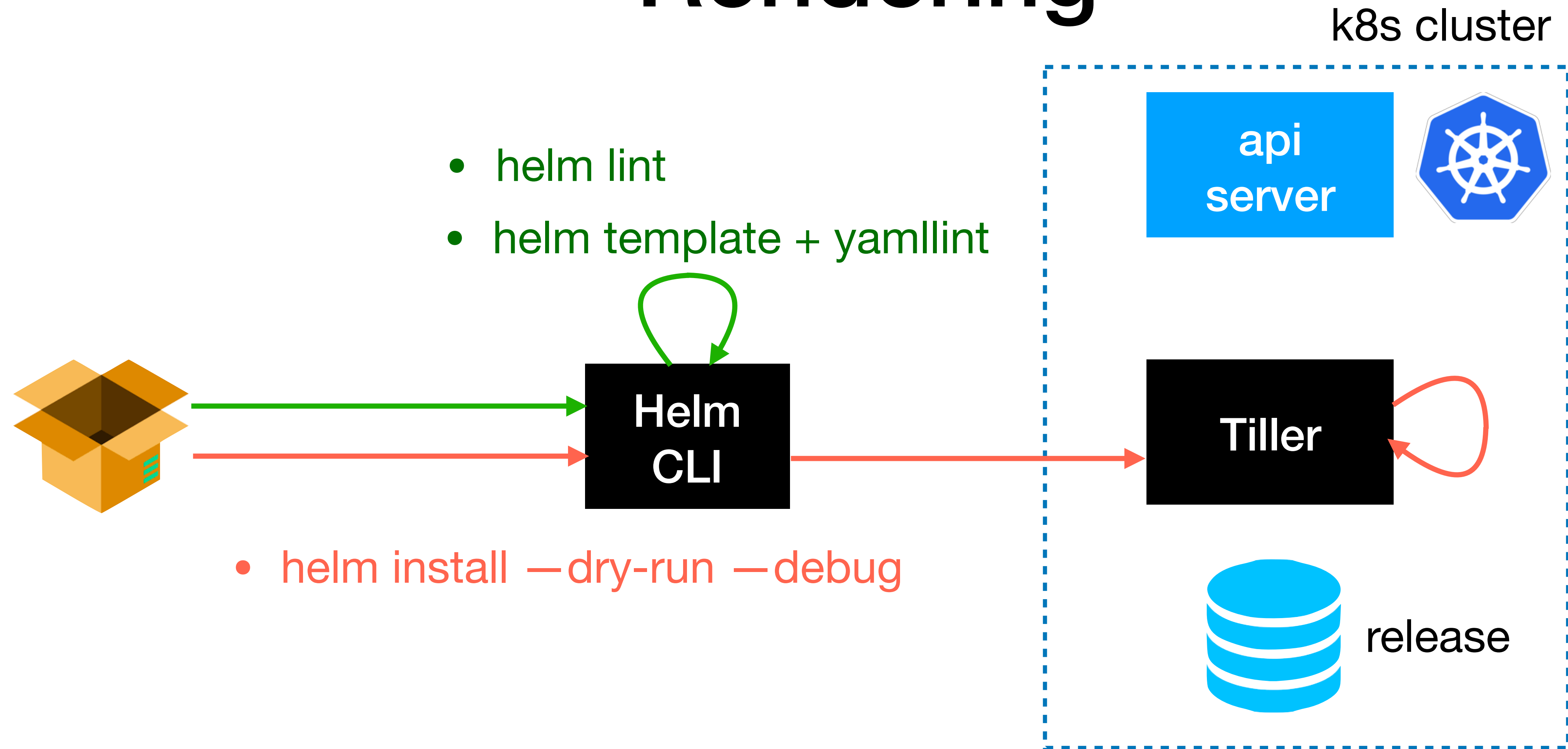


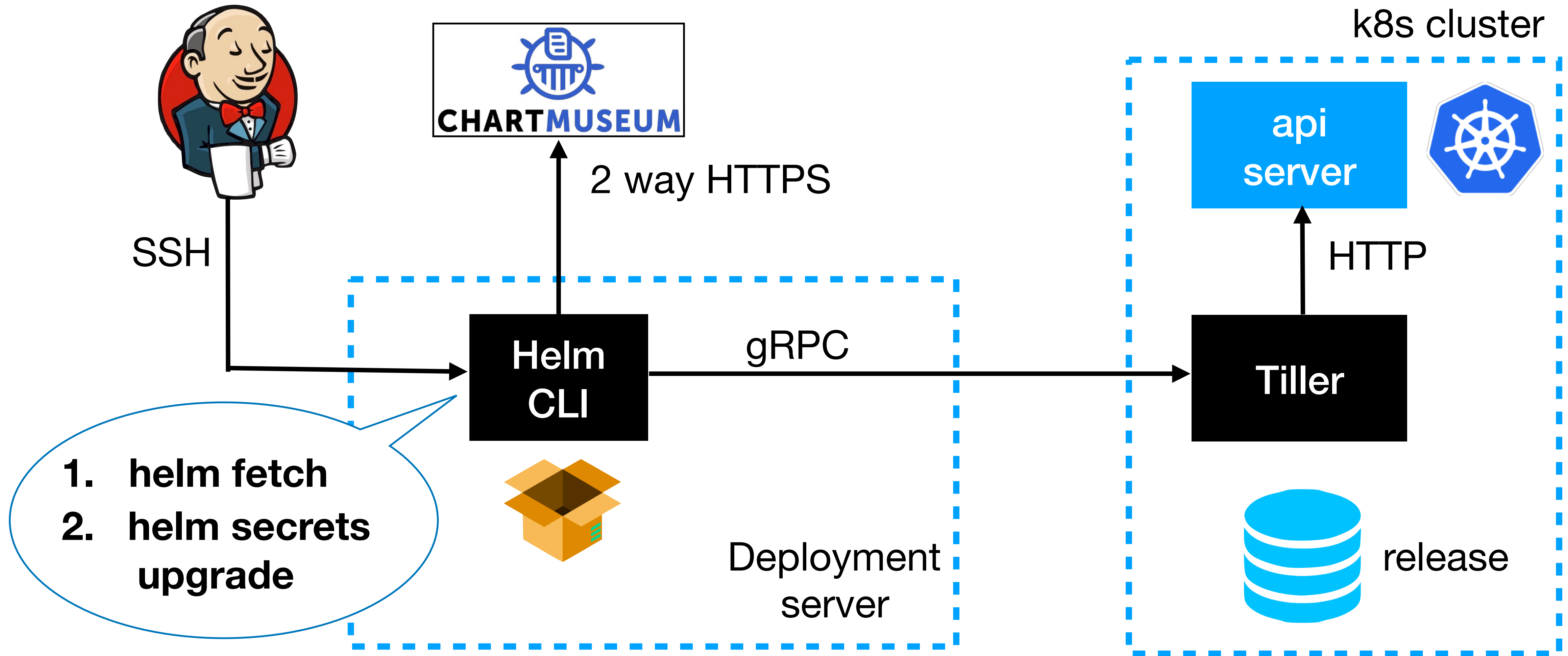
Chart Museum



Verify Template Rendering



Deployment



Deployment Status Check

- kubectl rollout status check with timeout
- Notification

Zoom Incoming Webhook robot **APP**

[Jenkins build status](#)

atom2-log-archiver/alpha#224 FAILED([Open](#))

Commit: ac6272973081dce8b7c25c069f04f8a41a12ad25

Image built: 1.1.4-1-gac62729

Zoom Incoming Webhook robot **APP**

[Jenkins build status](#)

atom2-log-archiver/alpha#225 SUCCESS([Open](#))

Commit: 4d8144fa22193437293a4b724e3683bf4f54f435

Image built: 1.1.4-2-g4d8144f

08/23/2019

Zoom Incoming Webhook robot **APP**

[Jenkins build status](#)

atom2-dgw/aws-iot#29 SUCCESS([Open](#))

Commit: e8494451c85ace10b4ae37d4293edc63a5661a50

Image built: feature.aws-iot.6-1-ge849445

Zoom Incoming Webhook robot **APP**

[Jenkins build status](#)

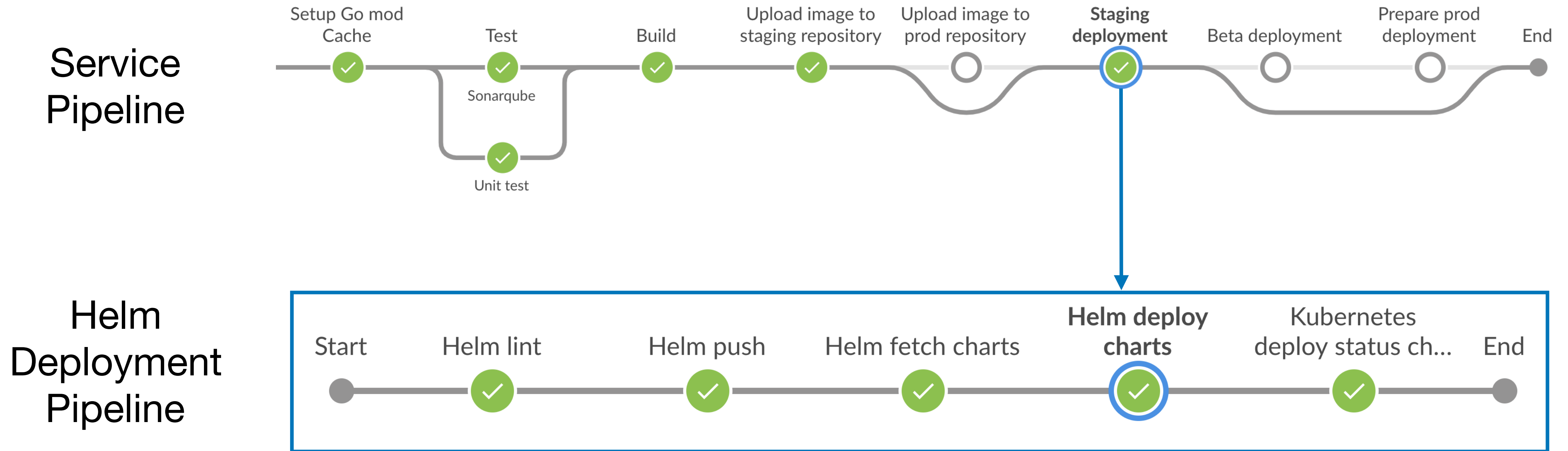
atom2-keymaster/int#47 SUCCESS([Open](#))

Commit: a9d481937ffa127b3da34ac1beb2a8f536c9d9c7

Image built: 0.0.7-7-ga9d4819

Reuse Jenkins Pipeline

- Apply **Parameter Jenkins Pipeline** if the service deployment steps are almost the same



- Why we need Helm?
- What is Helm?
- How do we use Helm?
- What's news in Helm?



Helm 3 Preview: Charting Our Future – Part 1: A History of Helm

Mon, Apr 22, 2019

On October 15th, 2015, the project now known as Helm was born. Only one year later, the Helm community joined the Kubernetes organization as Helm 2 was fast approaching. In June 2018, the Helm community [joined the CNCF](#) as an incubating project. Fast forward to today, and Helm 3 is nearing its first alpha release.

In this series of seven blog posts over the next four weeks, I'll provide some history on Helm's beginnings, illustrate how we got where we are today, showcase some of the new features available for the first alpha release of Helm 3, and explain how we move forward from here.

In order, I'll discuss:

1. The history of the creation of Helm
2. A Gentle Farewell to Tiller
3. Chart Repositories
4. Release Management
5. Changes to Chart Dependencies
6. Library Charts
7. What's Next?

Pre-release

v3.0.0-beta.3

5cb923e

Verified

Helm v3.0.0-beta.3



adamreese released this 5 days ago · [909 commits](#) to master since this release

Helm v3.0.0-beta.3 is the third beta release for Helm 3.

We want to take a moment to thank everyone that has been participating in finding issues, triaging, testing, and contributing fixes to the first beta release.

The community keeps growing, and we'd love to see you there!

- Join the discussion in [Kubernetes Slack](#):
 - `#helm-users` for questions and just to hang out
 - `#helm-dev` for discussing PRs, code, and bugs
- Hang out at the Public Developer Call: Thursday, 9:30 Pacific via [Zoom](#)
- Test, debug, and contribute charts: [GitHub/helm/charts](#)

Changelog

- fix(pkg/cli): do not override users xdg directories [b478848](#) (Adam Reese)
- Unify environment variable naming and use [1ea53d8](#) (Matt Farina)
- Displaying environment variables in alphanum order for env cmd [9191d10](#) (Matt Farina)
- Remove ability to have duplicates in environment variables [378b9dd](#) (Matt Farina)

Helm 3

Developer friendly, and
more secure to operate

Developer Friendly

- A single registry can be used to host both images and charts
- Built-in diff and push commands
- Library chart
- Compatible with Helm 2
- `helm delete --purge` is worked as default
- ...

More Secure to Operate

- No tiller
- Use k8s secret as default
- Improve upgrade strategy
- ...

3-way Strategic Merge Patches



```
$ helm upgrade myapp ./myapp
```

```
$ kubectl scale --replicas=0 deployment/myapp
```



```
$ helm rollback myapp --to-revision=0
```

In Helm 3, the patch is generated by using the old manifest, the new manifest and the live state

Any Impact ?

 [futuresimple](#) / [helm-secrets](#)

 Watch ▾ 33

[Code](#) [Issues 47](#) [Pull requests 7](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#)

Support helm v3 #130

 Open

kzap opened this issue on 26 Jul · 0 comments



kzap commented on 26 Jul

+ 😊 ...

Currently cant use Helm 3.0.0-alpha.2 with `helm secrets install`

I get this error

```
$ helm secrets install RELEASE-NAME CHART -f ./values.yaml
Error: unknown flag: --client
helm install: invalid option -- f
Error: plugin "secrets" exited with error
```

I believe this is because the format of `helm install` has changed in Helm V3 where there is no more `--name` and the release name is always required.



3

Conclusion

GitOps

- Use Git as the source of truth for both your infrastructure and application code
- Operation by pull request
- Trace all changes in Git
- End-to-end automation workflow
- Robust GitOps pipeline enhances developer experience

- Helm2 是很棒的工具，但要接受它有一些小缺點
- 期待 Helm3 可以讓部署服務在k8s上更接近 GitOps 的目標！



Thank You :)