

Customizing Charts with Helm Templates



Philippe Collignon

FREELANCE DEVOPS

@phcollignon www.phico.io



Helm Templates



Why Helm Templates ?

How does Helm Template engine work ?

When does Template engine run ?

Playing with Helm Template Data

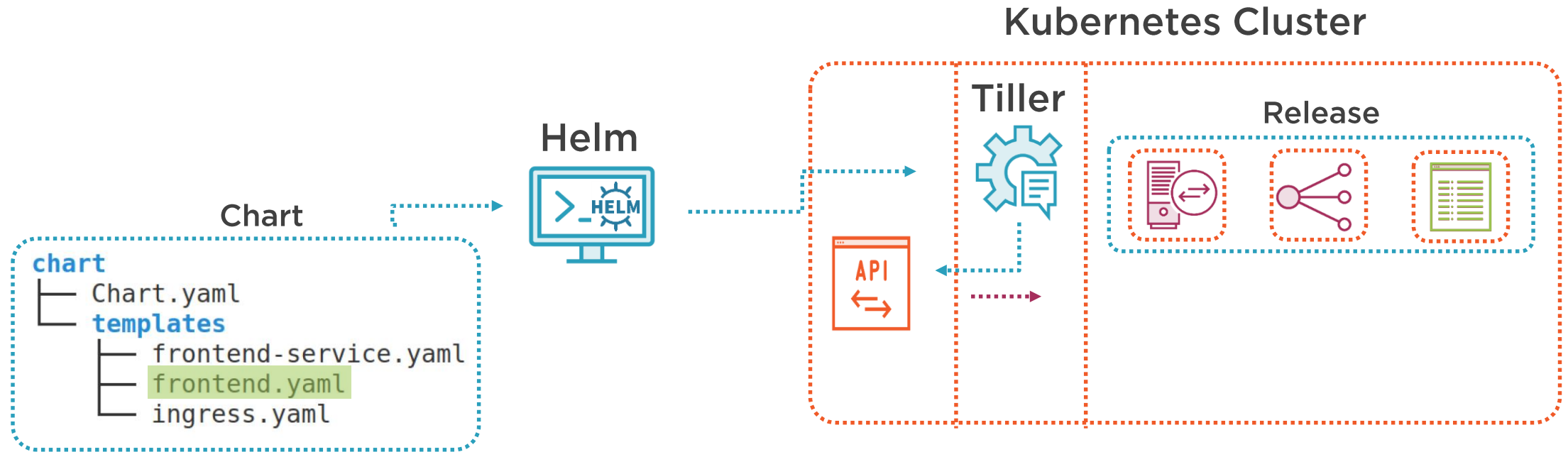
Adding Logic to Helm Template



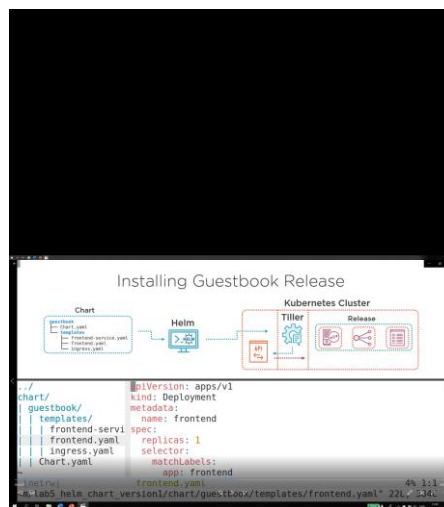
Why Helm Templates ?



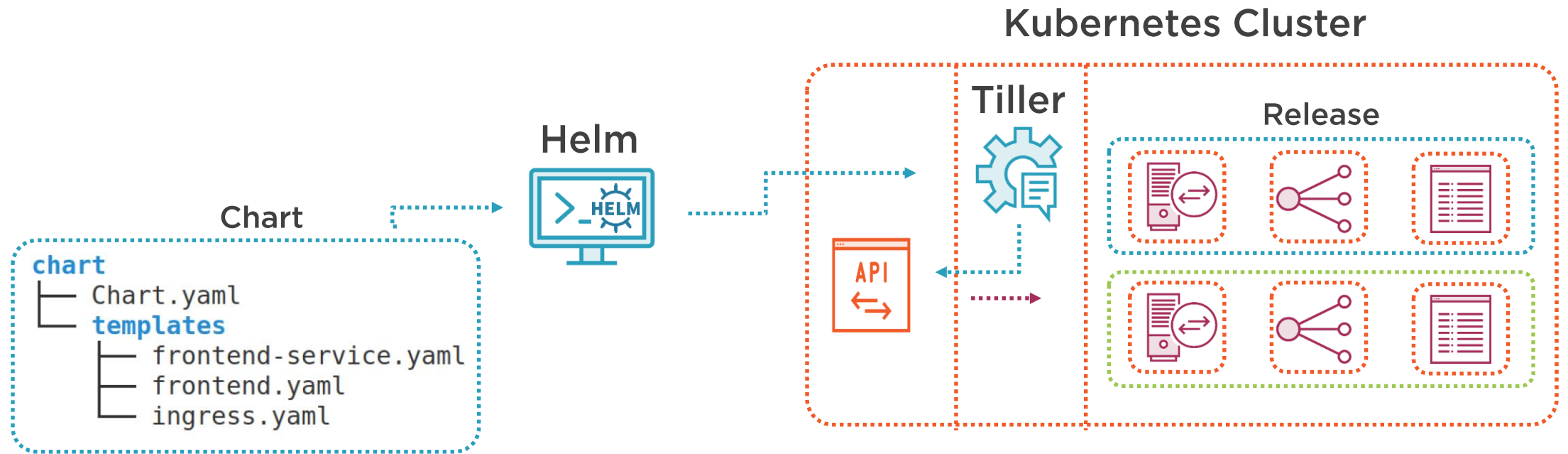
Release revision



Release revision



Helm Release



Helm Releases

"dev" Release

```
apiVersion: v1
kind: Service
metadata:
  labels:
    name: dev-frontend
  name: dev-frontend
spec:
  type: ClusterIP
  ports:
    - port: 80
      targetPort: 3000
  selector:
    app: dev-frontend
```

"test" Release

```
apiVersion: v1
kind: Service
metadata:
  labels:
    name: test-frontend
  name: test-frontend
spec:
  type: ClusterIP
  ports:
    - port: 80
      targetPort: 3000
  selector:
    app: test-frontend
```



Helm Template Engine



Analogy with other template engines

	Template engine	Directives
System {	bash	<code>\${ENV_VAR}</code>
	bat	<code>%ENV_VAR%</code>
Web dev {	Php	<code><?php echo \$title ?></code>
	Jsp, Asp, EJS	<code><%= title %></code>
Code generator {	Velocity	<code>\$var</code>
	Yeoman	<code><%= var %></code>
	Go Templates	<code>{{ var }}</code>



Go Template Engine

Template

```
apiVersion: v1
kind: Service
metadata:
  name: {{.name}}
```

```
type: NodePort
name: myservice
port: 80
```

Object



Manifest

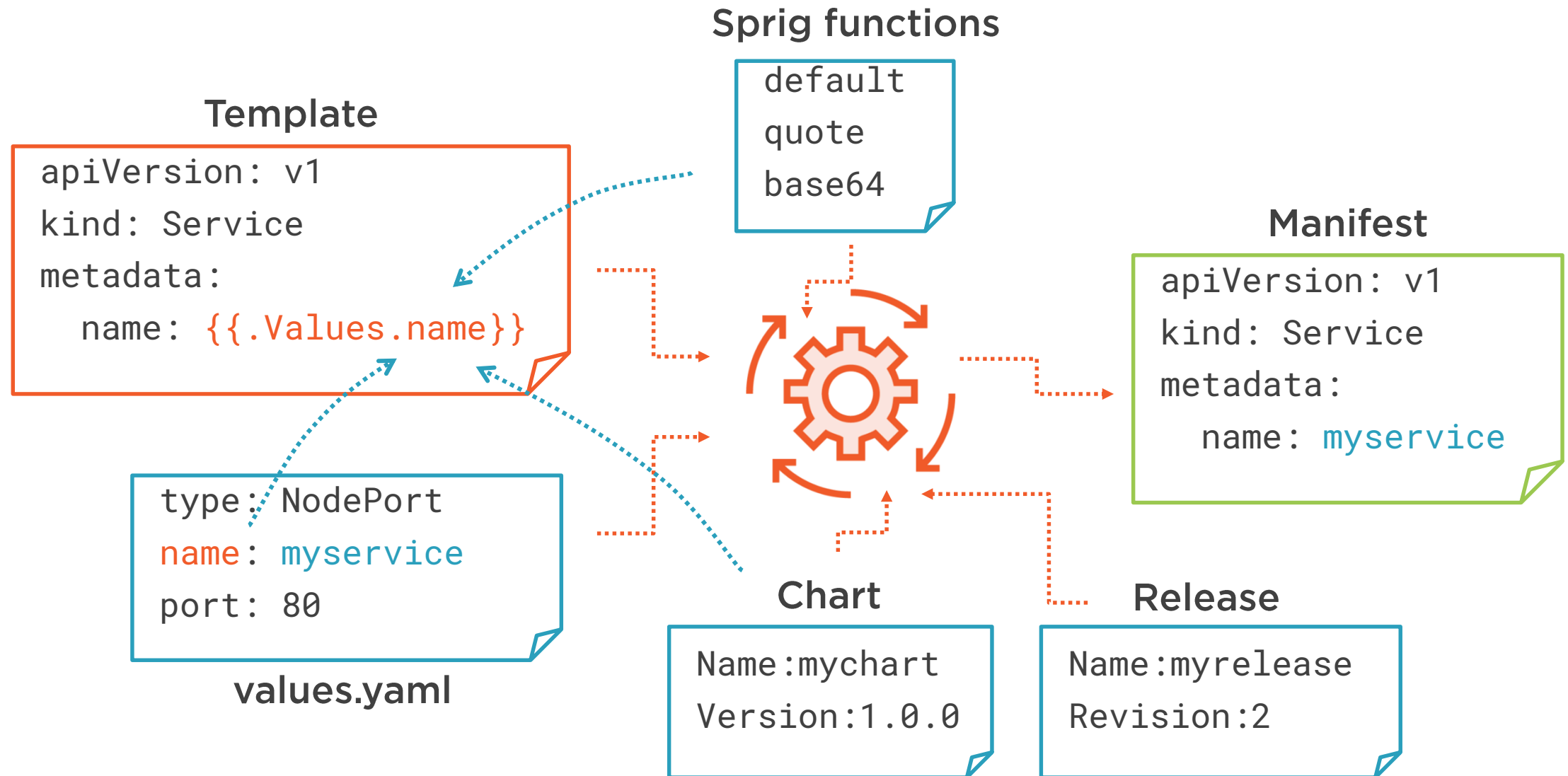
```
apiVersion: v1
kind: Service
metadata:
  name: myservice
```



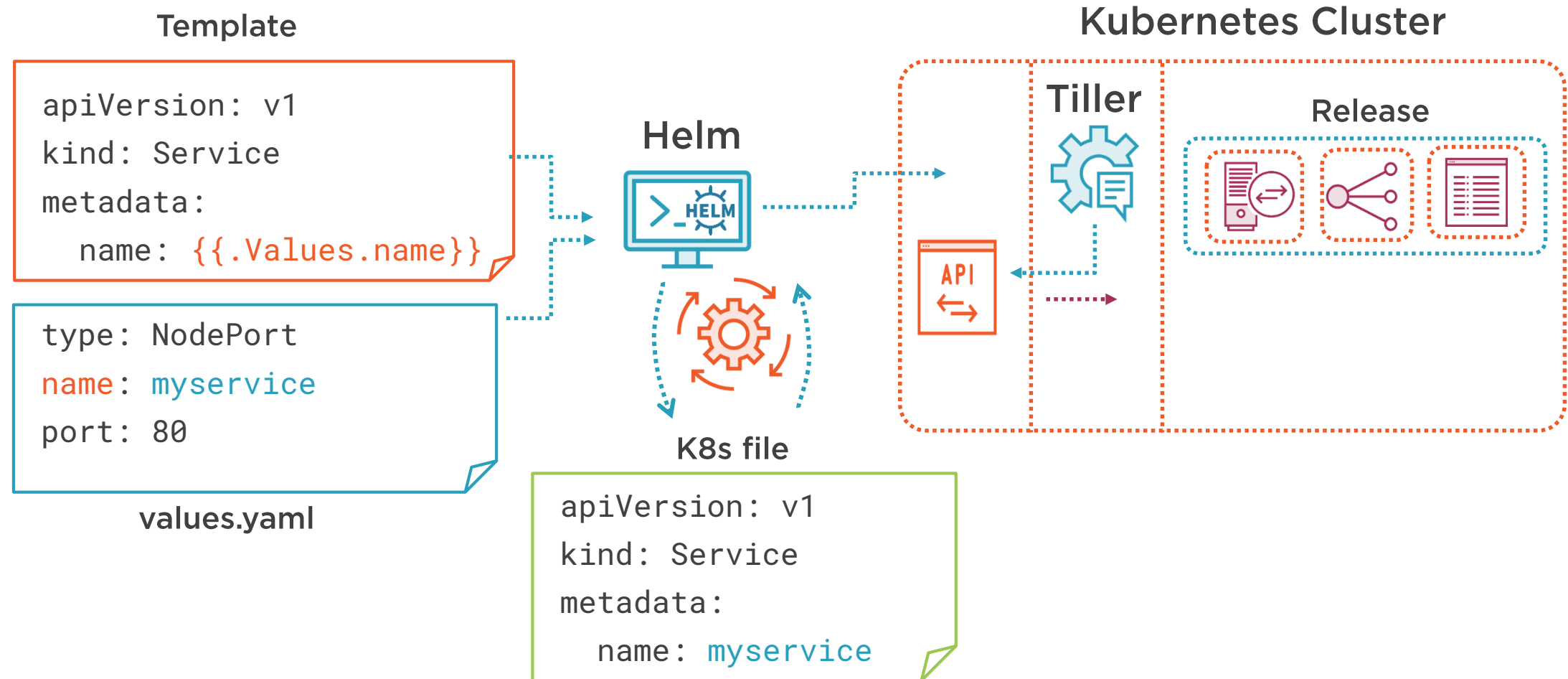
<https://github.com/phcollignon/Go-Template>



Helm Template Engine



Helm Template Execution



hardy-moose.v1 - Kubernetes

127.0.0.1:37663/api/v1/namespaces/kube-system/services/http:kubernetes-dashboard:/proxy/#/configmap/kube-system/hardy-moose.v1?nam...

kubernetes

Search

+ CREATE

Config and storage > Config Maps > hardy-moose.v1

EDITDELETE

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Discovery and Load Balancing

Ingresses

Services

Config and Storage

Config Maps

Persistent Volume Claims

Secrets

Settings

Details

Name:

hardy-moose.v1

Namespace:

kube-system

Labels:

MODIFIED_AT: 1560011651

NAME: hardy-moose

OWNER: TILLER

STATUS: DELETED

VERSION: 1

Creation Time:

2019-06-08T15:56 UTC

Data

release:

H4sIAAAAAAAC/+Ra248cR9XXzsqyXV8ePgYEyBK6GAIFtNzM463kUX24l0Wey6ZXXttI7Sq6a7ZaW/f0l09zsTep0jkNUhc3hBSXojCK+IFKW8x8MQLSPwFSKAEv4QnJFB1dfVU93TP9Fw2duR5mumqOnX0r06f8zunBvxfH7nasGTatoeL66Bwr1B84dyvf/uvv5/5/ydv//JXhQuX9IL5956TH/97c0P3lm59LktbGCi2xZUbdMxMMEXfr4Bvgv0H/nYI13bPpb0V0WaXL304jr8PjZMqPaRS2DPduE0nwJrcqVTGFTvr9bkyoWPvgDwwdkuUo+xpFHV35iweoNNhVUupSpXiu8UwBcJNh0DEeyVD/vYcLDrycQxip+sPHxYghru6RaGUriR3PMNw0Im1mDp5ATQGxPyreR4WMvGmwNs0vqGo7mZE54BInrWyq8Ug++6uae3+vpb0CpNNoAG95IkuPqFulB6aJXuuhJU05gAyMPy01kYihvUrvZ91ySLW3se/G9VfCVESsh3SUPqy4m8hCZRvHtVeTo7Hr6balwEEVH0uWpsC9YAowMUEaIkgBEFJbFfjwIdQt1fC1VBxleHISioeALzRt68jWuiXf1YP1D3TSj1AMdQnnHPquHmg0xz7UrpcFyC56ykXv1QA6Cp4USlDKZQnKvoddqhGEsoM874HtahB+PQGw1mWuJSm1VyrVV8qR/34P+aS/Z/uuiq8hzdQtCX4LPoLdK5expcJH8HXfJhimaMmxPzkBL43bPcFkelb/KIAX087KHegqZof1uJBxWMGc2GkZqIsNTw1UzHtyMxwy8BysUulk60CEdUxh0sBk0rZLQk2oz9vEVm1DgdL+ZlsKIaRTUqXQgRHSBL1HmLSdyfVKpQIg9LCBVWK7TD5ynFzaF5+cEWMFnxA/JczIsjIcbxyhPQWdgx7aGJrvlcjQs3FjqGryIvZHD7ctH0rtDlunImI2r8pnGtueyE9qNDaUJagPf0YMBH5gaRzuVH0o9owQbqF3UhwCeom0or7SPCE2mt70rHdITw5UcaGCToS37HgYds3jLZt60pQgevGAzT0ovEZnHzkc4LFpVBC

Test Helm template before Helm install

Static

`helm template`

works without tiller

static release-name

Dynamic

`helm install --dry-run --debug`

real helm install but without commit

generated release-name

```
> helm template mychart
```

```
> helm template mychart -x templates/service.yaml
```

```
> helm install mychart --debug --dry-run
```

```
> helm install mychart --debug --dry-run 2>&1 | less
```

Playing with Helm Template Data



Helm Template Values

values.yaml

other-file.yaml

helm install -f file

variables

helm install - - set foo=bar

**Hierarchically
organized**

Template

```
apiVersion: v1
kind: Service
metadata:
  name: {{.Values.service.name}}
```

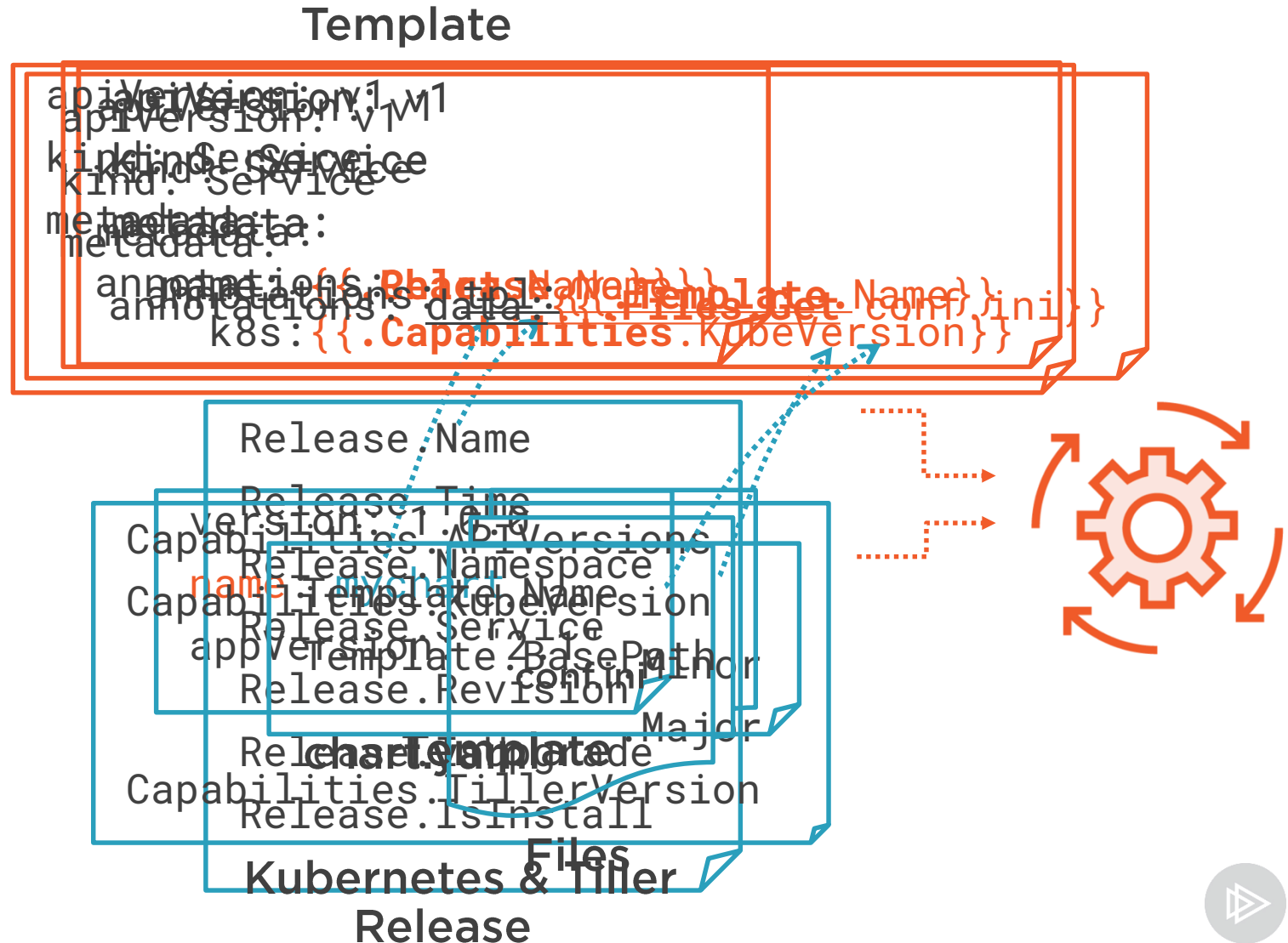
```
service:
  type: NodePort
  name: myservice
  port: 80
```

values.yaml

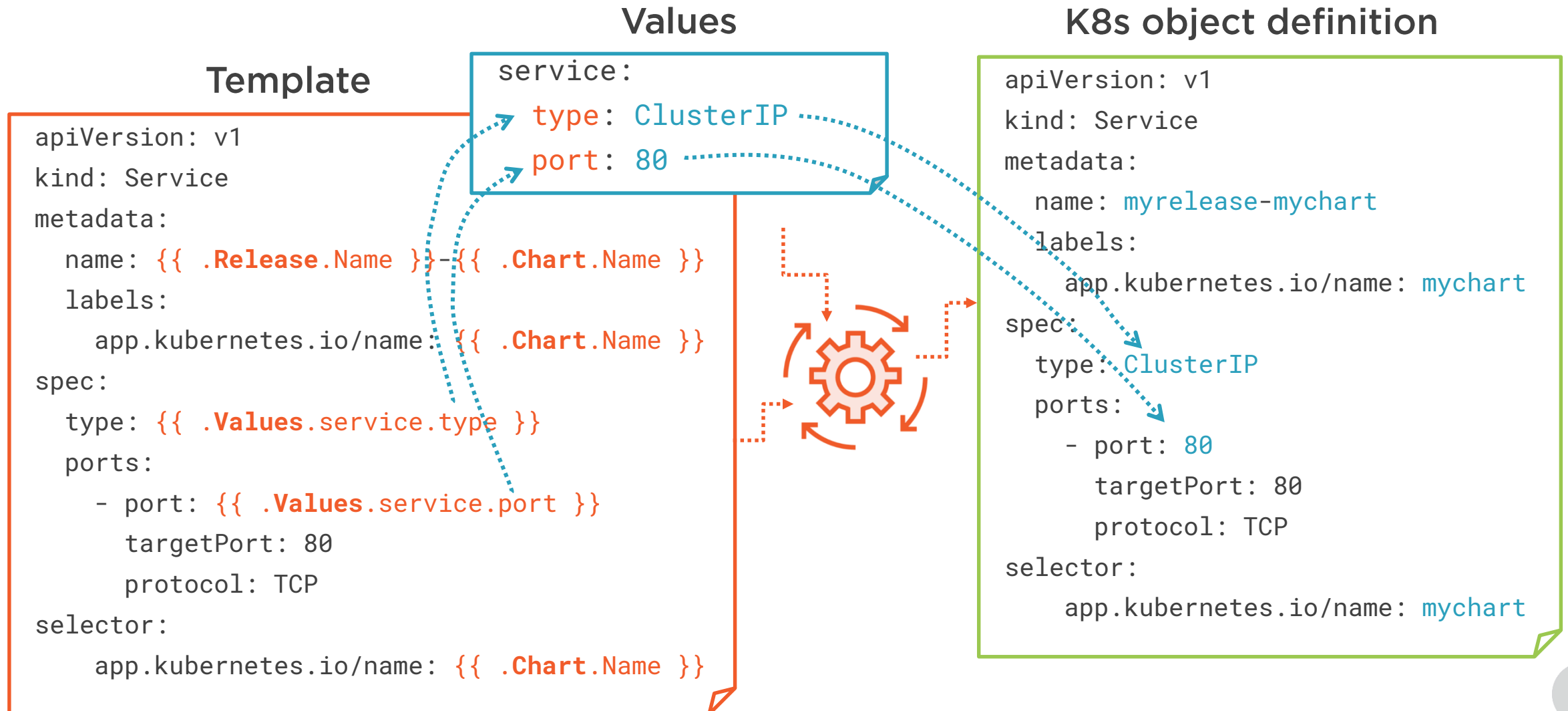


Helm Template Built-in Objects

Chart data
Release data
Kubernetes data
File data
Template data



Helm Template Example



Values and sub charts

```
guestbook/  
| charts/  
| | backend/  
| | | templates/  
| | | Chart.yaml  
| | | values.yaml  
| | database/  
| | frontend/  
| Chart.yaml  
| values.yaml
```

```
secret:  
  mongodb_uri:  
    username: your_user  
    password: your_password  
service:  
  type: NodePort
```

Child Chart
values.yaml

```
Guestbook:  
  title: "Concert for Climate"  
  backend:  
    secret:  
      mongodb_uri:  
        username: admin  
        password: password
```

Parent Chart
values.yaml



COMPUTED VALUES:

backend:

global: {}

image:

repository: phico/backend

tag: "2.0"

ingress:

enabled: false

host: backend.minikube.local

replicaCount: 1

secret:

mongodb_uri:

dbchart: your_db_chart

password: password

username: admin

service:

port: 80

type: ClusterIP

database:

global: {}

secret:

mongodb_password: password

mongodb_username: admin

service:

port: 80

:■

Global values

```
guestbook/  
| charts/  
| | backend/  
| | | templates/  
| | | Chart.yaml  
| | | values.yaml  
| | database/  
| | frontend/  
| Chart.yaml  
| values.yaml
```

```
secret:  
  mongodb_uri:  
    username: your_user  
    password: your_password
```

Child Chart
values.yaml

```
service:  
  type: ClusterIP  
  global:  
    id: "Id<AcB123FDx23>"
```

```
metadata:  
  labels:  
    app.id : {{ .Values.global.id }}
```

Parent Chart
values.yaml

```
username: admin  
password: password
```



Adding Logic to Helm Template



Adding Logic to Helm Template

Using Functions & Pipelines

Modifying Scope using "with"

Controlling whitespace & indent

Logical operators in Helm Templates

Flow control (conditions & loops)

Using variables

Calling Helper functions & Sub-templates



Using Functions and Pipelines



Functions and Pipeline

Function

quote value

quote(value)

default default_value value

default (default_value, value)

Pipeline

value | quote



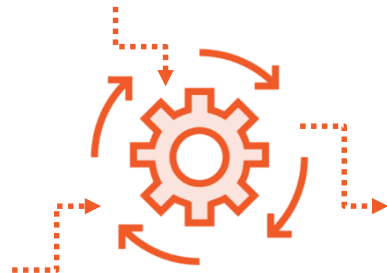
value | default default_value



default

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: {{ .Values.service.name | default .Chart.Name }}
```

```
service:
  type: NodePort
  name: myservice
  port: 80
```



```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: myservice
```



Functions and Pipelines



Go text/template package (<http://golang.org/pkg/text/template/>)



Sprig functions project (<http://masterminds.github.io/sprig/>)

Helm project (<http://helm.sh>)



No Go custom functions support (use helper functions)

Functions and Pipelines

Function

default default_value value

quote value

upper value

trunc value 63

trimSuffix "-" value

b64enc value

randAlphaNum 10

toYaml value

printf format value ...

Pipeline

value | default default_value

value | quote

value | upper

value | trunc 63

value | trimSuffix "-"

value | b64enc

value | randAlphaNum 10

value | toYaml

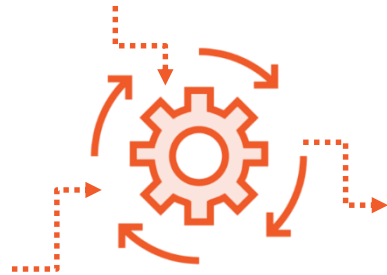
list value ... | join " -"



trunc & trimSuffix

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: {{ .Values.service.name | trunc 63 | trimSuffix "-" }}
```

```
service:
  type: NodePort
  name: myservice-
  port: 80
```

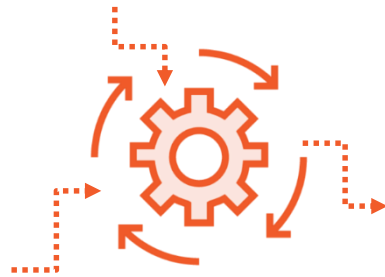


```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: myservice
```



b64enc & quote

```
apiVersion: v1
kind: Secret
[...]
data:
  mongodb-root-password: {{ .Values.mongodbRootPassword | b64enc | quote }}
```



```
mongodbRootPassword : T9rYGFAMGE
```

```
apiVersion: v1
kind: Secret
[...]
data:
  mongodb-root-password: "VD1yWUdGQU1HRQ=="
```



Modifying Scope using "With"



Modifying scope using With

```
spec:  
  type: {{ .Values.service.type }}  
  ports:  
    - port: {{ .Values.service.port }}  
      targetPort: 80
```



```
spec:  
  {{ with .Values.service }}  
  type: {{ .type }}  
  ports:  
    - port: {{ .port }}  
      targetPort: 80  
  {{ end }}
```

```
service:  
  type: ClusterIP  
  port: 80
```



```
spec:  
  type: ClusterIP  
  ports:  
    - port: 80  
      targetPort: 80
```



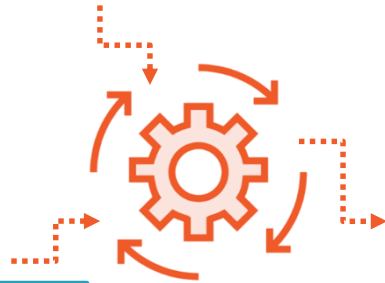
Controlling whitespaces and indent



Removing extra carriage returns

spec:

```
{{- with .Values.service }}  
type: {{ .type }}  
ports:  
  - port: {{ .port }}  
    targetPort: 80  
{{- end -}}
```



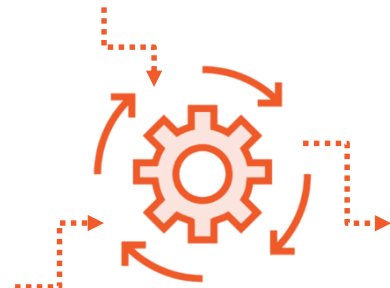
```
service:  
  type: ClusterIP  
  port: 80
```

```
spec:  
  type: ClusterIP  
  ports:  
    - port: 80  
      targetPort: 80
```

Removing extra carriage returns

```
spec:
  type: ClusterIP
  ports:
    - port:
{{- with .Values.service -}}
  {{ .port }}
{{- end -}}
  targetPort: 80
```

```
service:
  port: 80
```

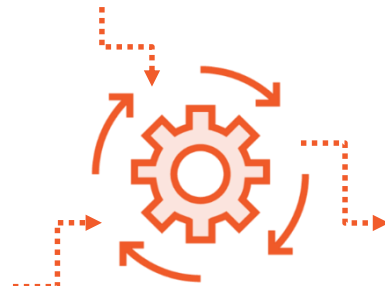


```
spec:
  type: ClusterIP
  ports:
    - port: 80targetPort: 80
      targetPort: 80
      80
      targetPort: 80
```

Indent by Template

```
spec:
  type: ClusterIP
  ports:
    - port: 80
      targetPort: 80
    {{ indent 6 tcpValues.tcp }}
```

```
service:
  port: 80
  tcp: "protocol: TCP"
```

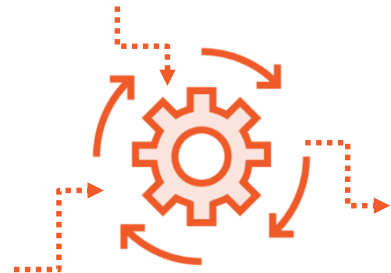


```
spec:
  type: ClusterIP
  ports:
    - port: 80
      targetPort: 80
  protocol: TCP
```

Printing with format

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: {{- printf "%s-%s" .Release.Name .Values.service.name -}}
```

```
service:
  type: NodePort
  name: myservice
  port: 80
```



```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: myrelease-myservice
```

Logical operators



Logical operators are functions

Operator	Function	Example
Equal to	eq	or .Val1 .Val2
Not equal to	ne	ne .Val1 .Val2
Greater than	gt	gt .Val1 .Val2
Lower than	lt	lt .Val1 .Val2
Or	or	or .Val1 .Val2
And	and	and .Val1 .Val2
Not	not	not .Val1



Example of logical operators

```
{{- if and .adminEmail (or .serviceAccountJson  
.existingSecret) }}
```

(Cf. oauth2-proxy : deployment.yaml)

```
{{- if (and (eq .Values.service.type "NodePort") (not (empty  
.Values.service.nfsNodePort))) }}
```

(Cf. nfs-server-provisioner : service.yaml)

```
{{- if or .Values.rbac.pspEnabled (and .Values.rbac.namespaced  
(or .Values.sidecar.dashboards.enabled  
.Values.sidecar.datasources.enabled)) }}
```

(Cf. grafana : role.yaml)



Flow control



Conditions

```
{{- if Values.ingress.enabled -}}  
apiVersion: extensions/v1beta1  
kind: Ingress  
metadata:  
  labels:  
    app.kubernetes.io/name:  
{{- if .Values.service.name -}}  
{{ .Values.service.name | trimSuffix "-" }}  
{{- else -}}  
{{ .Chart.Name }}  
{{- end -}}  
{{- end -}}
```

```
ingress:  
  enabled: true  
  name: myingress
```



```
apiVersion: extensions/v1beta1  
kind: Ingress  
metadata:  
  labels:  
    app.kubernetes.io/name: myingress
```

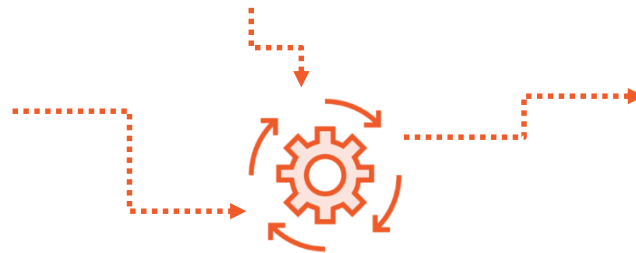


Loops

```
apiVersion: extensions/v1beta1
kind: Ingress
[...]
spec:
  {{- range .Values.ingress.hosts }}
    - host: {{ .hostname | quote }}
      http:
        paths:
          {{- range .paths }}
            - path: {{ .path }}
              backend:
                serviceName: {{ .service }}
                servicePort: http
          {{- end }}
        {{- end }}
```

```
ingress:
  hosts:
    - hostname: frontend.local
      paths:
        - path: "/public"
          service: "frontend"
        - path: "/admin"
          service: "admin"
    - hostname: backend.local
      paths: []
```

```
apiVersion: extensions/v1beta1
kind: Ingress
[...]
spec:
  rules:
    - host: "frontend.local"
      http:
        paths:
          - path: /public
            backend:
              serviceName: frontend
              servicePort: http
          - path: /admin
            backend:
              serviceName: admin
              servicePort: http
    - host: "backend.local"
      http:
        paths:
```



Using variables



Why variables ?

```
spec:
  {{ with .Values.service }}
  type: {{ .type }}
  ports:
    - port: {{ .port }}
      targetPort: 80
      name {{ .Values.defaultPortName }}
  {{ end }}
```



Why variables ?

```
apiVersion: extensions/v1beta1
kind: Ingress
[...]
spec:
  {{- range .Values.ingress.hosts }}
    - host: {{ .hostname | quote }}
      http:
        paths:
          {{- range .paths }}
            - path: {{ .path }}-{{ .hostname }}
              backend:
                serviceName: {{ .Release.Name }} {{ .service }}
                servicePort: http
          {{- end }}
        {{- end }}
  {{- end }}
```



Using a variable

```
{{ $defaultPortName := .Values.defaultPortName }}
spec:
  {{ with .Values.service }}
  type: {{ .type }}
  ports:
    - port: {{ .port }}
      targetPort: 80
      name {{ $defaultPortName }}
  {{ end }}
```



Using variable and Global variable

```
apiVersion: extensions/v1beta1
kind: Ingress
[...]
spec:
  {{- range $currentHost := values $.ingress.hosts }}
  {{- $currentHost.hostname | quote }}
  - host: {{ .hostname | quote }}
    httpPaths:
      {{range .paths }}
      {{- path := .path }}
      - path: {{ path }}-{{ $currentHost.suffix }}
        backend:
          serviceName: {{ $.Release.Name }} {{ .service }}
          servicePort: http
      {{- end }}
    {{- end }}
  {{- end }}
```



Calling Helper functions and Sub-templates



Why using helper functions

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/name: {{- if .Values.fullnameOverride -}}
    {{- .Values.fullnameOverride | trunc 63 | trimSuffix "-" -}}
    {{- else -}}
    {{- printf "%s-%s" .Release.Name $name | trunc 63 |
    trimSuffix "-" -}}
    {{- end -}}
```

```
service:
  type: NodePort
  name: myservice
  port: 80
```

```
metadata:
  labels:
    app.kubernetes.io/name: myservice
```




Using a sub-template

`_helpers.tpl`

```
{{- define "mychart.fullname" -}}  
{{- if .Values.fullnameOverride -}}  
{{- .Values.fullnameOverride | trunc 63 | trimSuffix "-" -}}  
{{- else -}}  
{{- printf "%s-%s" .Chart.Name .Release.Name | trunc 63 | trimSuffix "-" -}}  
{{- end -}}  
{{- end -}}
```

```
apiVersion: v1  
kind: Deployment  
metadata:  
  labels:  
    app.kubernetes.io/name: {{ include "mychart.fullname" . }}  
  [...]   
spec:  
  selector:  
    matchLabels:  
      app.kubernetes.io/name: {{ include "mychart.fullname" . }}
```



_helpers.tpl

```
mychart/  
| charts/  
| templates/  
| | tests/  
| | NOTES.txt  
| | _helpers.tpl  
| | deployment.yaml  
| | ingress.yaml  
| | service.yaml  
| .helmignore  
| Chart.yaml  
| values.yaml
```

Where is _helpers.tpl file ?

Files prefixed by '_' are not rendered as Kubernetes objects

.helmignore



Indenting a sub-template

```
apiVersion: v1
kind: Deployment
metadata:
  labels:
    app.kubernetes.io/name: {{ include "mychart.fullname" . }}
    {{ include "mychart.myfunnylabel" . }}
    {{ template "mychart.myfunnylabel" . }}
  [...]
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: {{ include "mychart.fullname" . }}
```



NOTES.txt

1. Get the application URL by running these commands:

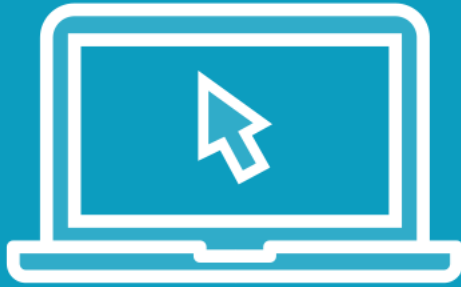
```
{{- if .Values.ingress.enabled }}
{{- range $host := .Values.ingress.hosts }}
  {{- range .paths }}
    http{{ if $.Values.ingress.tls }}s{{ end }}://{{ $host.host }}{{ .path }}
  {{- end }}
{{- end }}
```

NOTE: It may take a few minutes for the LoadBalancer IP to be available.

```
You can watch the status of by running 'kubectl get --namespace {{
.Release.Namespace }} svc -w {{ include "mychart.fullname" . }}'
export SERVICE_IP=$(kubectl get svc --namespace {{ .Release.Namespace }}
{{ include "mychart.fullname" . }} -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
echo http://$SERVICE_IP:{{ .Values.service.port }}
```



Demo

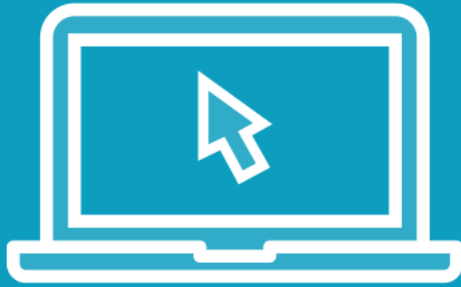


Customizing Guestbook Helm Chart : Frontend Chart values

```
http://github.com/phcollignon/helm/  
lab7_helm_template_begin  
lab7_helm_template_final
```



Demo

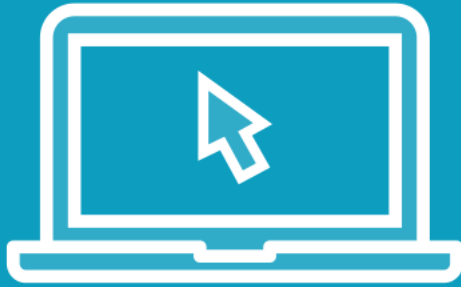


Customizing Guestbook Helm Chart : Adding logic

```
http://github.com/phcollignon/helm/  
lab8_helm_template_begin  
lab8_helm_template_final
```



Demo

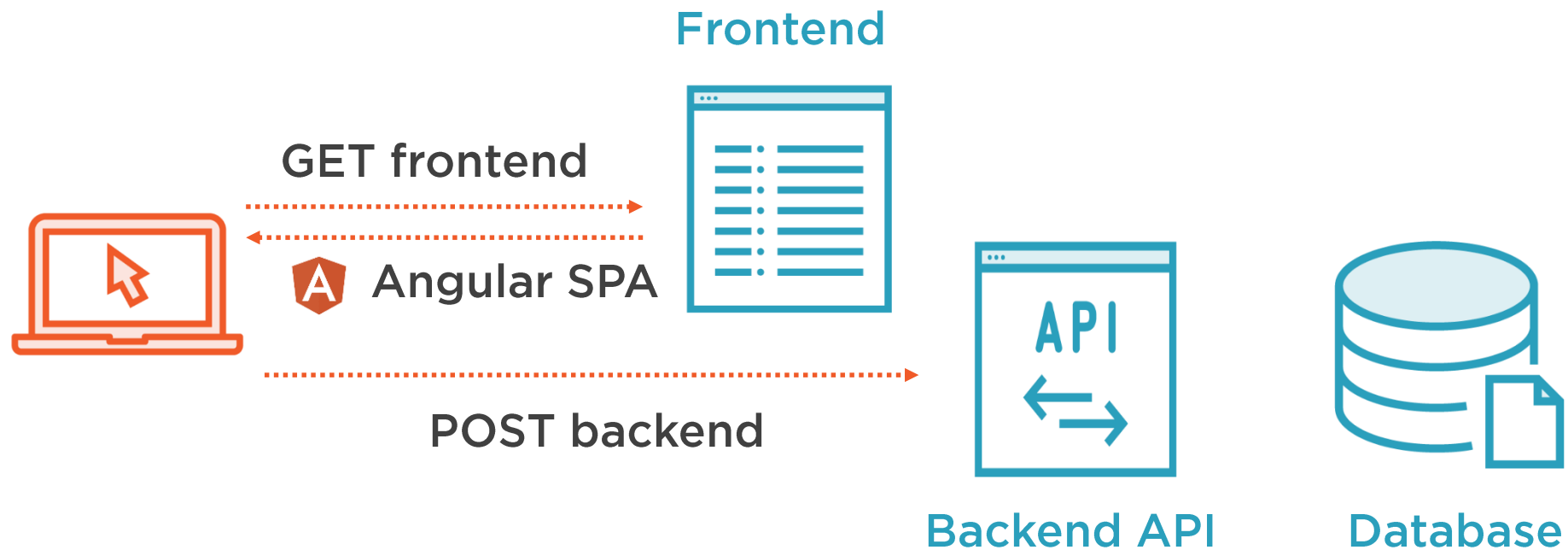


Customizing Guestbook Helm Chart : Installing Dev & Test Releases

```
http://github.com/phcollignon/helm/  
lab9_helm_template_begin  
lab9_helm_template_final
```



GuestBook application : Angular SPA + API



Helm Templates



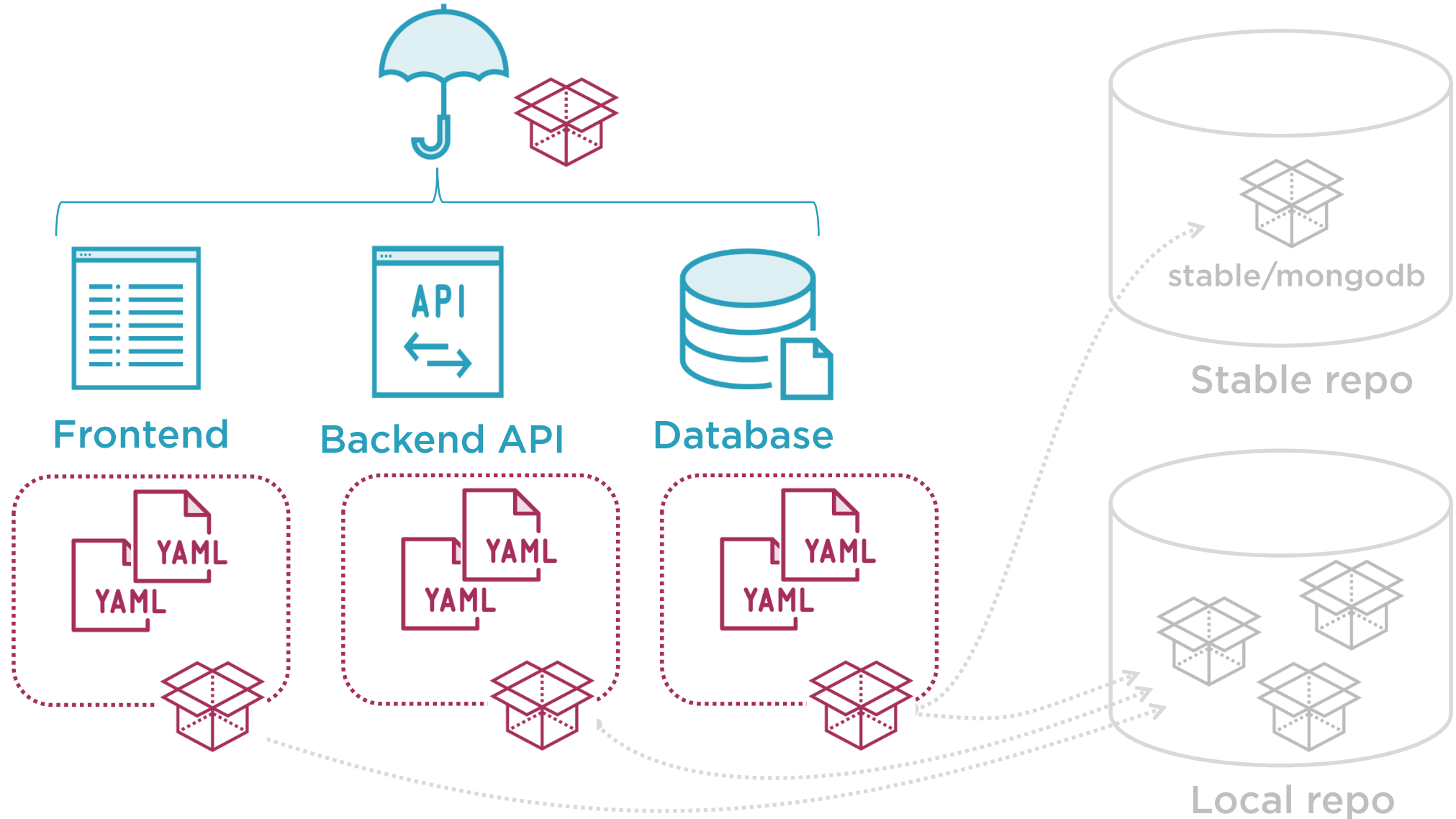
Helm Template engine ?

Helm Template Data

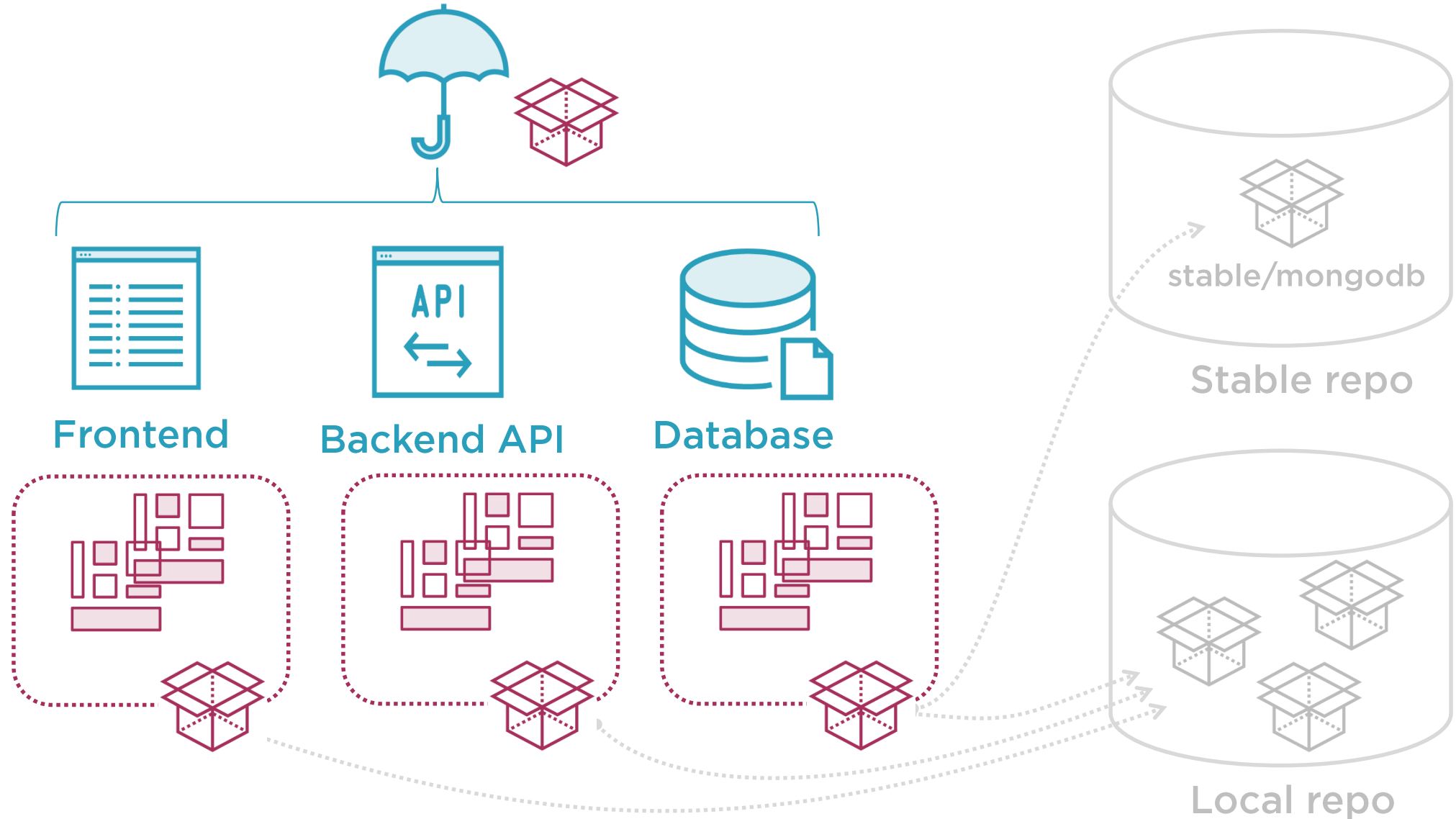
Adding Logic to Helm Template

- Functions and Pipelines
- Scope with “width”
- Controlling withespaces
- Logical operators
- Flow controls
- Variables
- Sub-templates

Guestbook application



Guestbook application



Guestbook application

