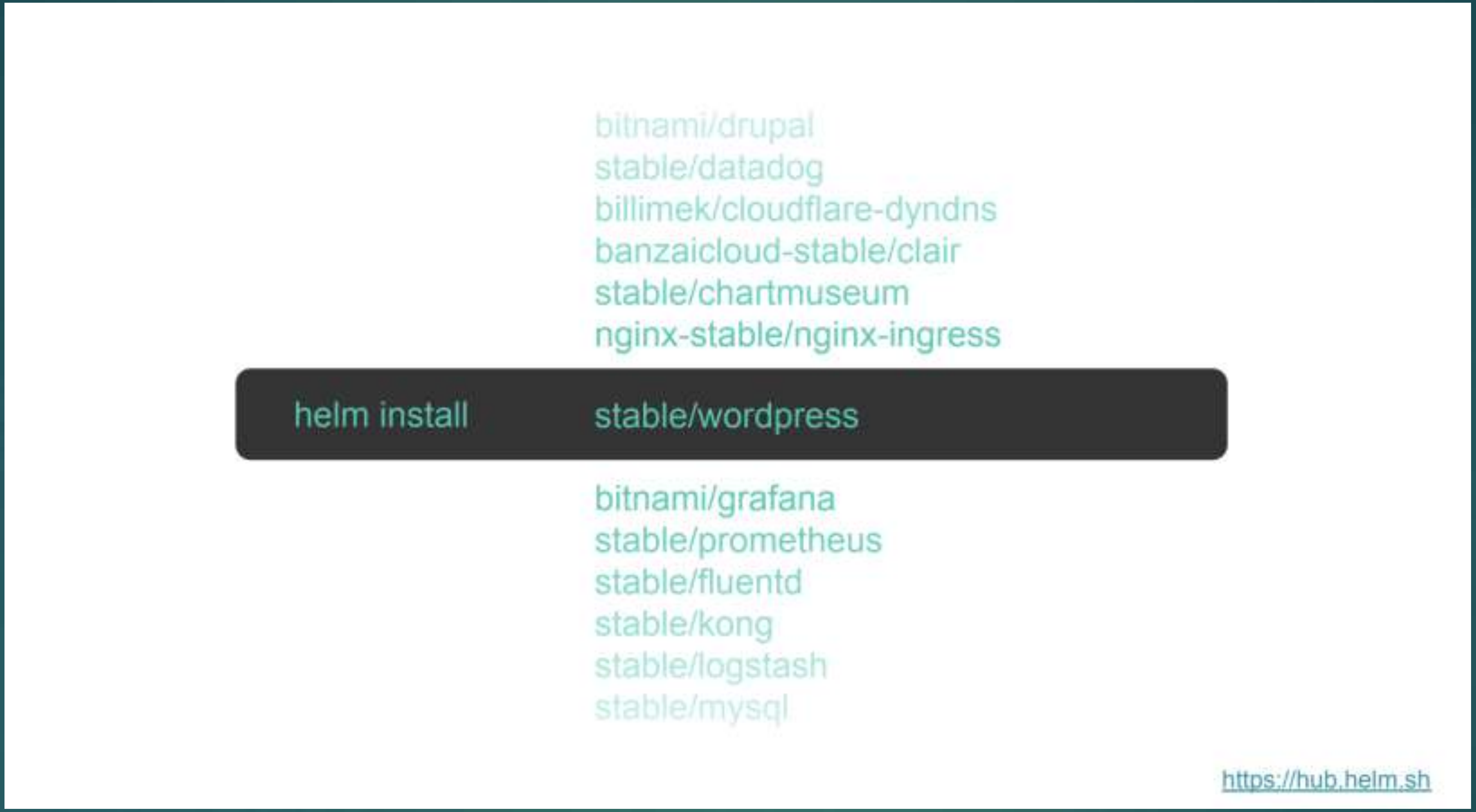
Itroduction To Helm

If Kubernetes is the operating system of the internet,

then Helm is a package manager for that



bitnami/drupal
stable/datadog
billimek/cloudflare-dyndns
banzaicloud-stable/clair
stable/chartmuseum
nginx-stable/nginx-ingress

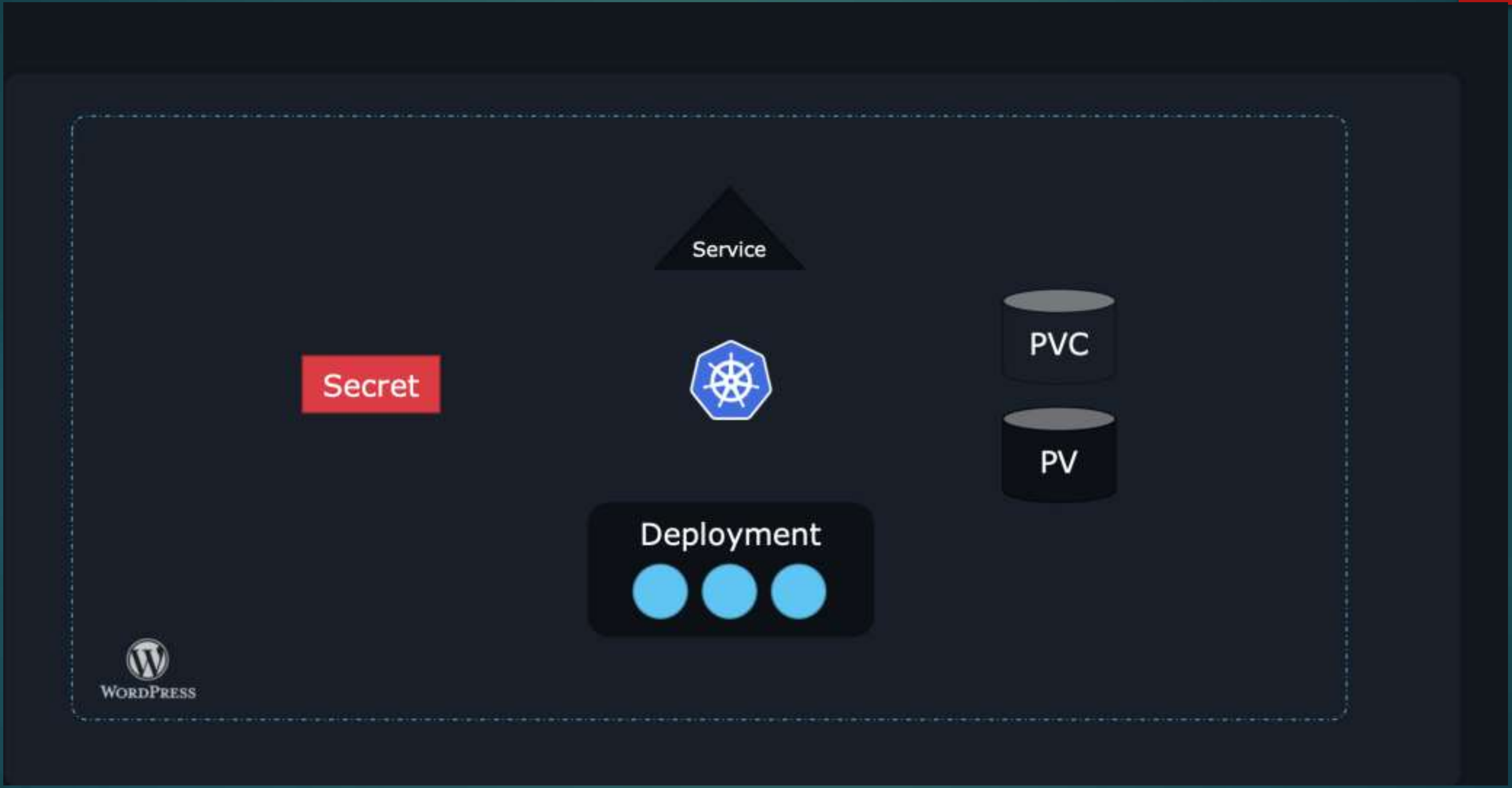
```
helm install        stable/wordpress
```

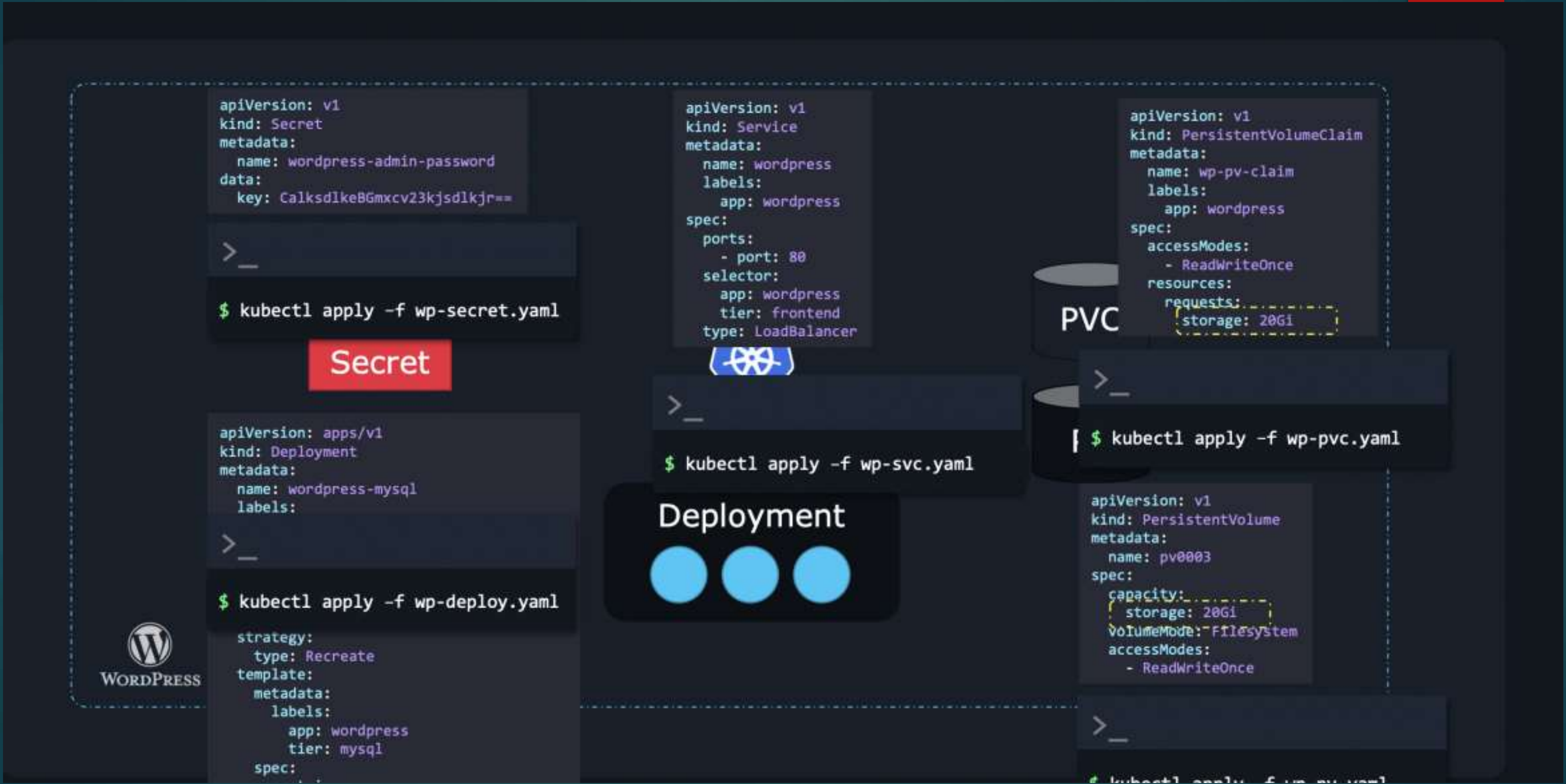
bitnami/grafana
stable/prometheus
stable/fluentd
stable/kong
stable/logstash
stable/mysql

https://hub.helm.sh

What is Helm?

Normal WordPress App Depenecies in K8s chaleenge





```yaml
apiVersion: v1
kind: Secret
metadata:
  name: wordpress-admin-password
data:
  key: CalksdlkeBGmxcv23kjsdlkjr==
```

```
>_
$ kubectl apply -f wp-secret.yaml
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  ports:
    - port: 80
  selector:
    app: wordpress
    tier: frontend
  type: LoadBalancer
```

```
>_
$ kubectl apply -f wp-svc.yaml
```

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```

```
>_
$ kubectl apply -f wp-pvc.yaml
```

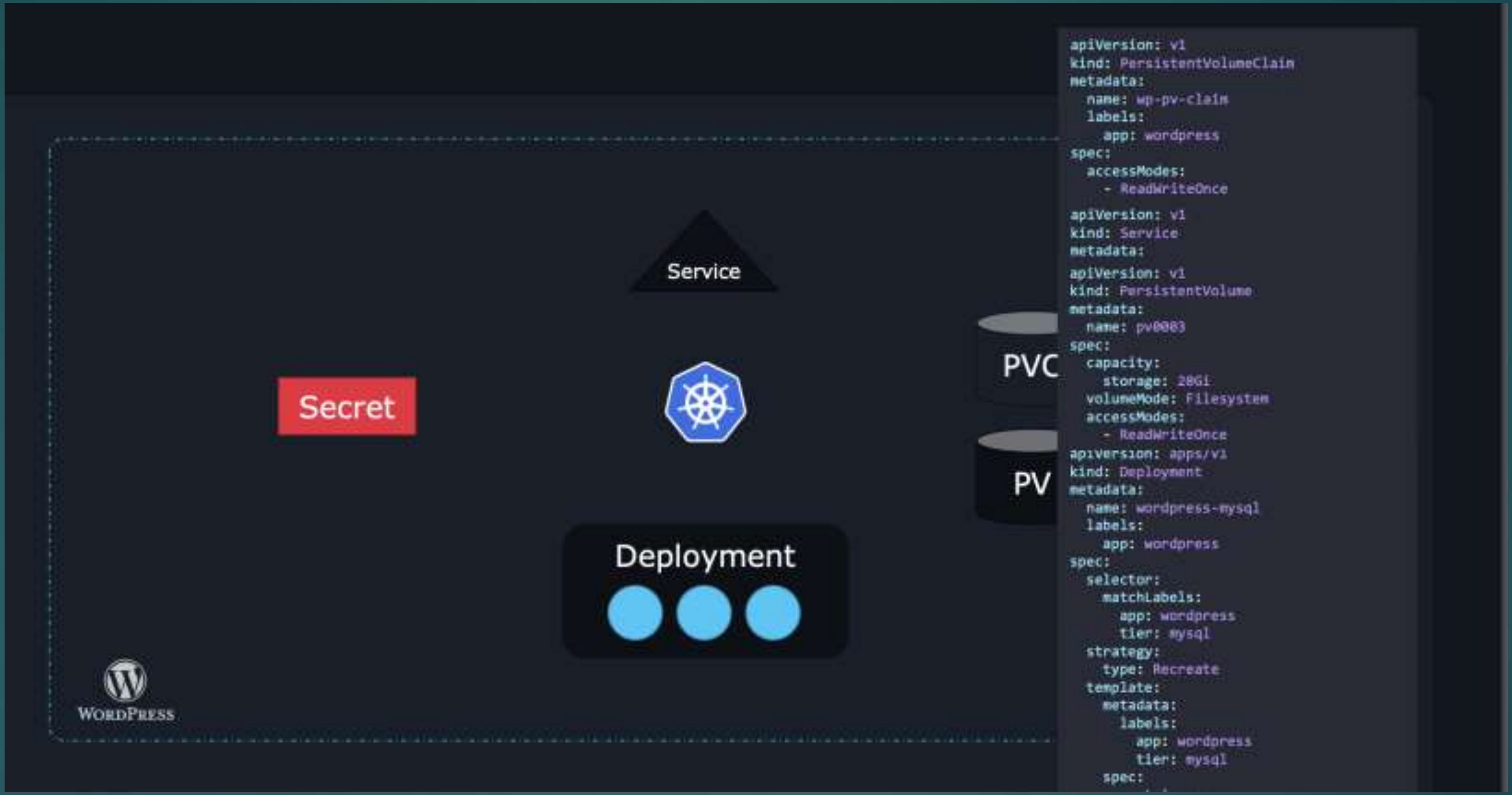
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
```

```
>_
$ kubectl apply -f wp-deploy.yaml
```

```yaml
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
    spec:
```

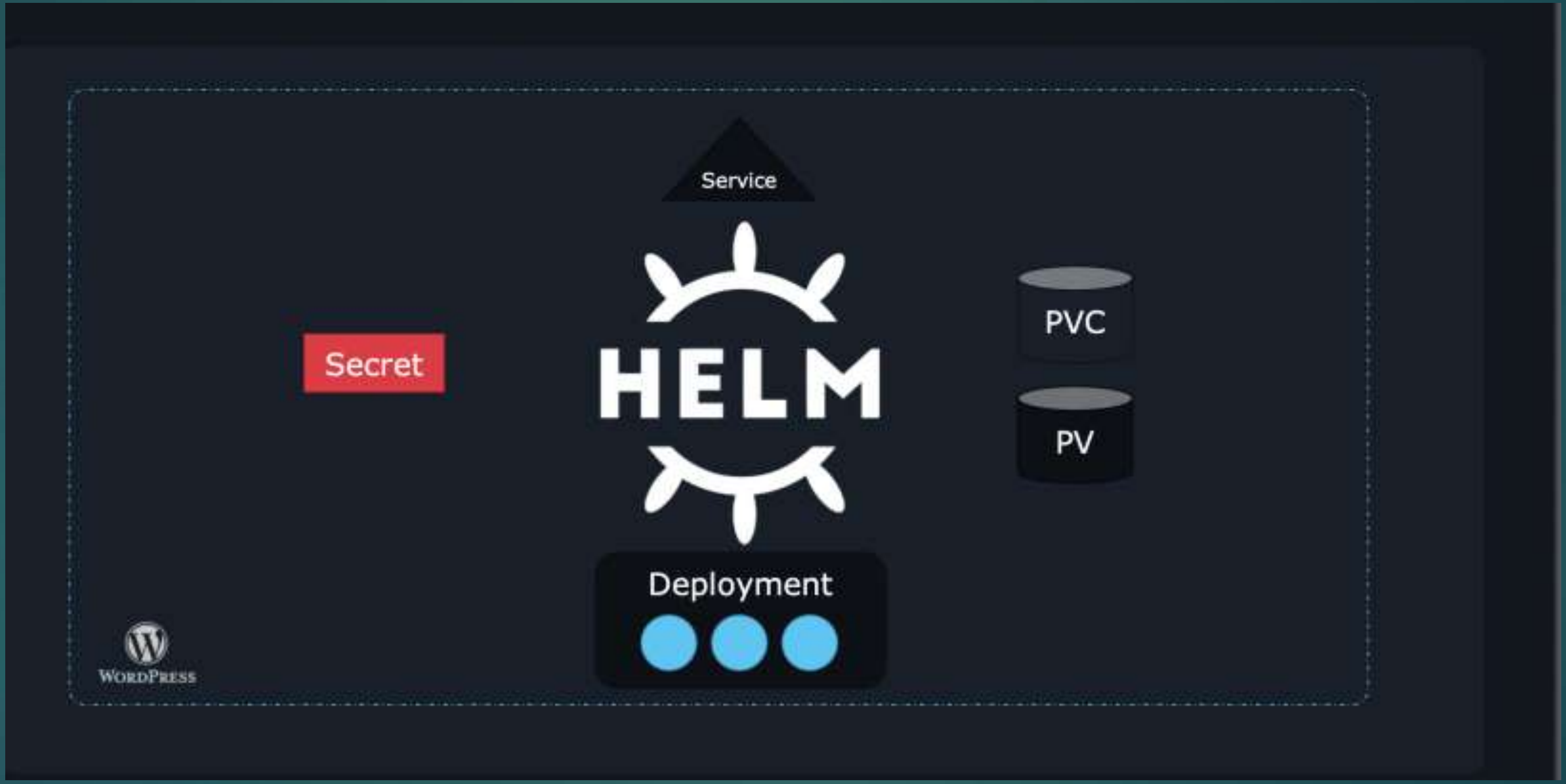
```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0003
spec:
  capacity:
    storage: 20Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
```

```
>_
$ kubectl apply -f wp-pv.yaml
```

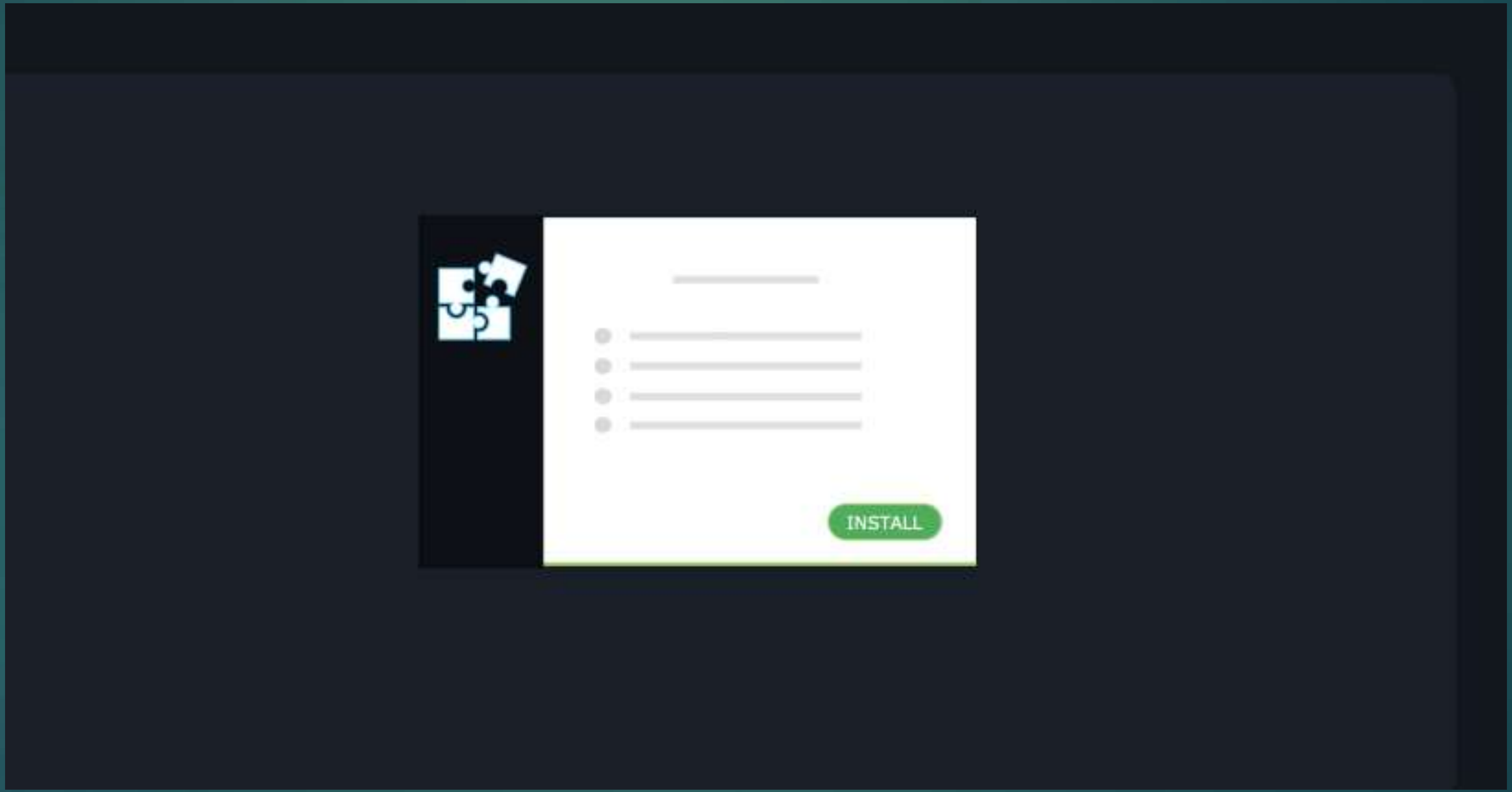


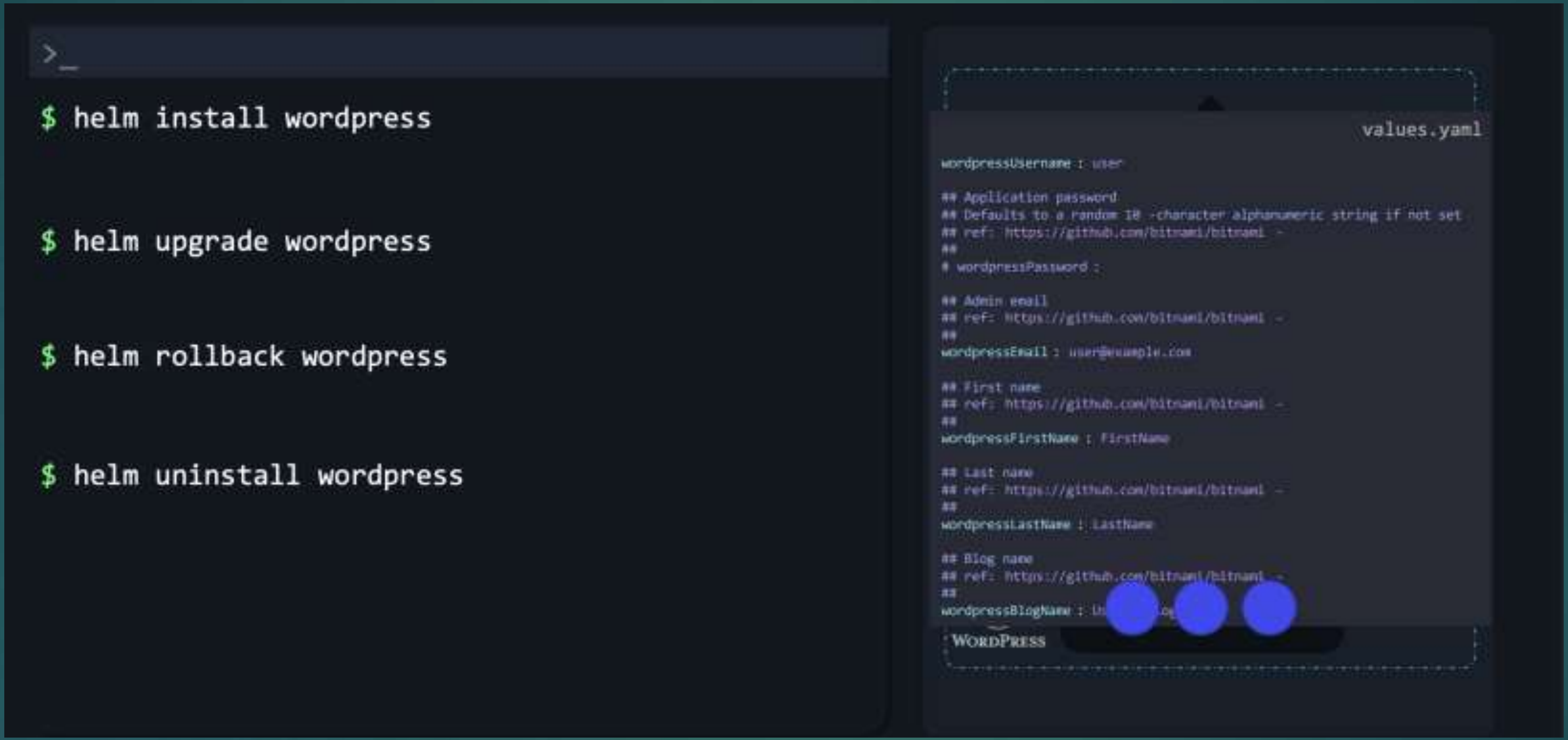
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
apiVersion: v1
kind: Service
metadata:
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0003
spec:
  capacity:
    storage: 20Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
apiversion: apps/v1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
      tier: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
    spec:
```

Here we solved the problem using Helm



1 APP



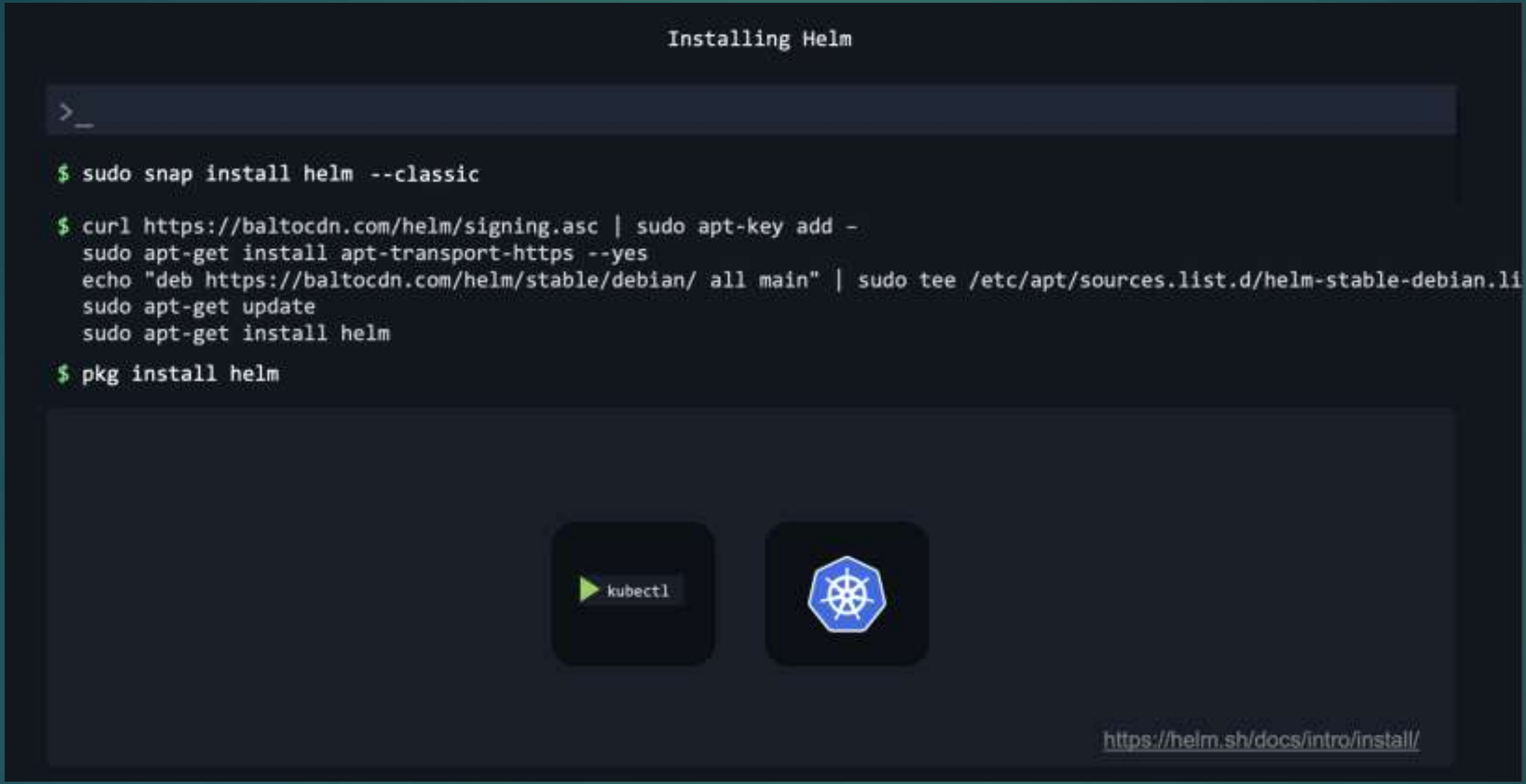


```
>_

$ helm install wordpress

$ helm upgrade wordpress

$ helm rollback wordpress

$ helm uninstall wordpress
```

values.yaml

```
wordpressUsername : user

## Application password
## Defaults to a random 10 -character alphanumeric string if not set
## ref: https://github.com/bitnami/bitnami -
##
# wordpressPassword :

## Admin email
## ref: https://github.com/bitnami/bitnami -
##
wordpressEmail : user@example.com

## First name
## ref: https://github.com/bitnami/bitnami -
##
wordpressFirstName : FirstName

## Last name
## ref: https://github.com/bitnami/bitnami -
##
wordpressLastName : LastName

## Blog name
## ref: https://github.com/bitnami/bitnami -
##
wordpressBlogName : Un[illegible]og
```

Installing HELM

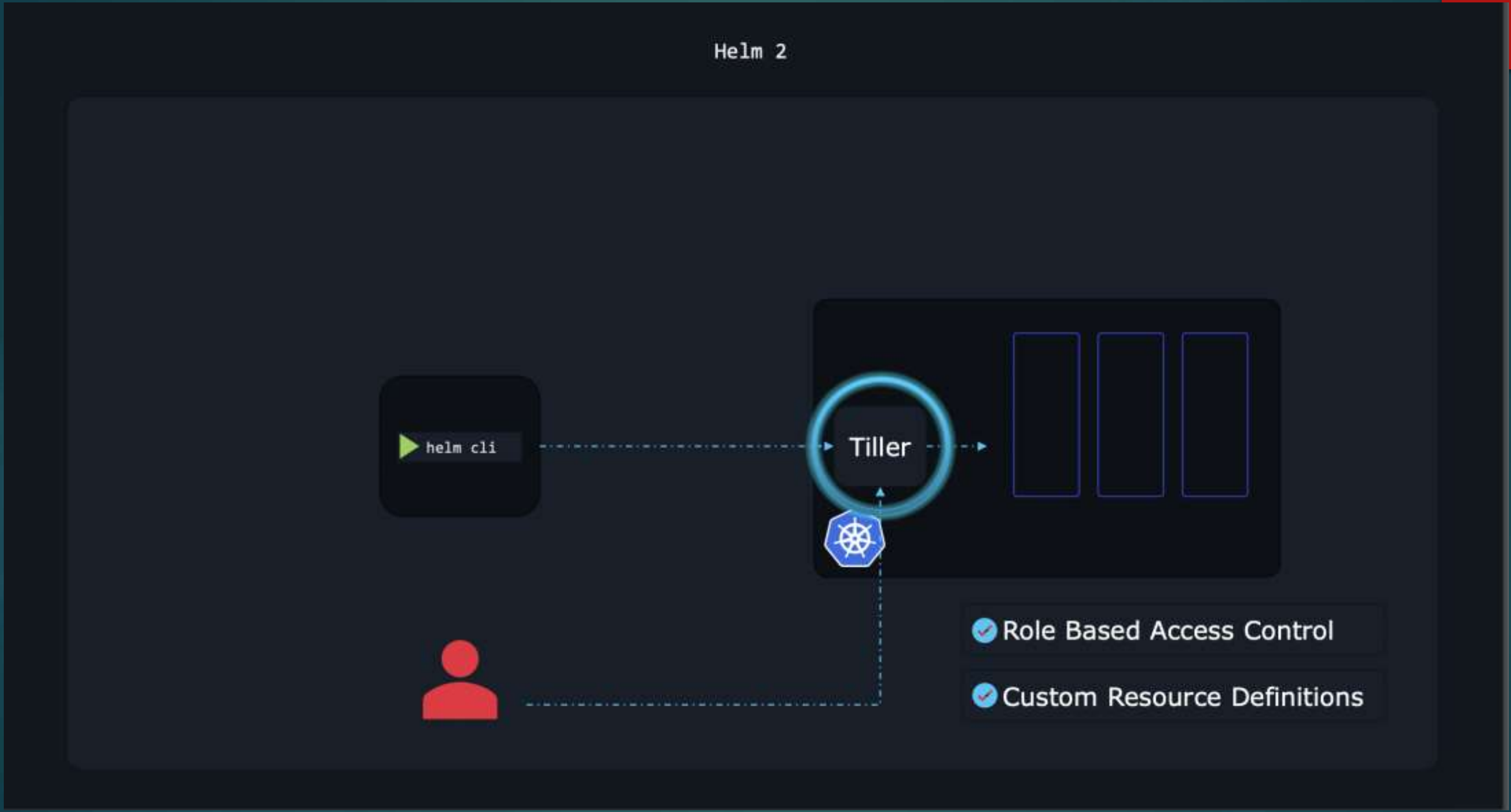


Installing Helm

```
>_

$ sudo snap install helm --classic

$ curl https://baltocdn.com/helm/signing.asc | sudo apt-key add -
  sudo apt-get install apt-transport-https --yes
  echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.li
  sudo apt-get update
  sudo apt-get install helm

$ pkg install helm
```

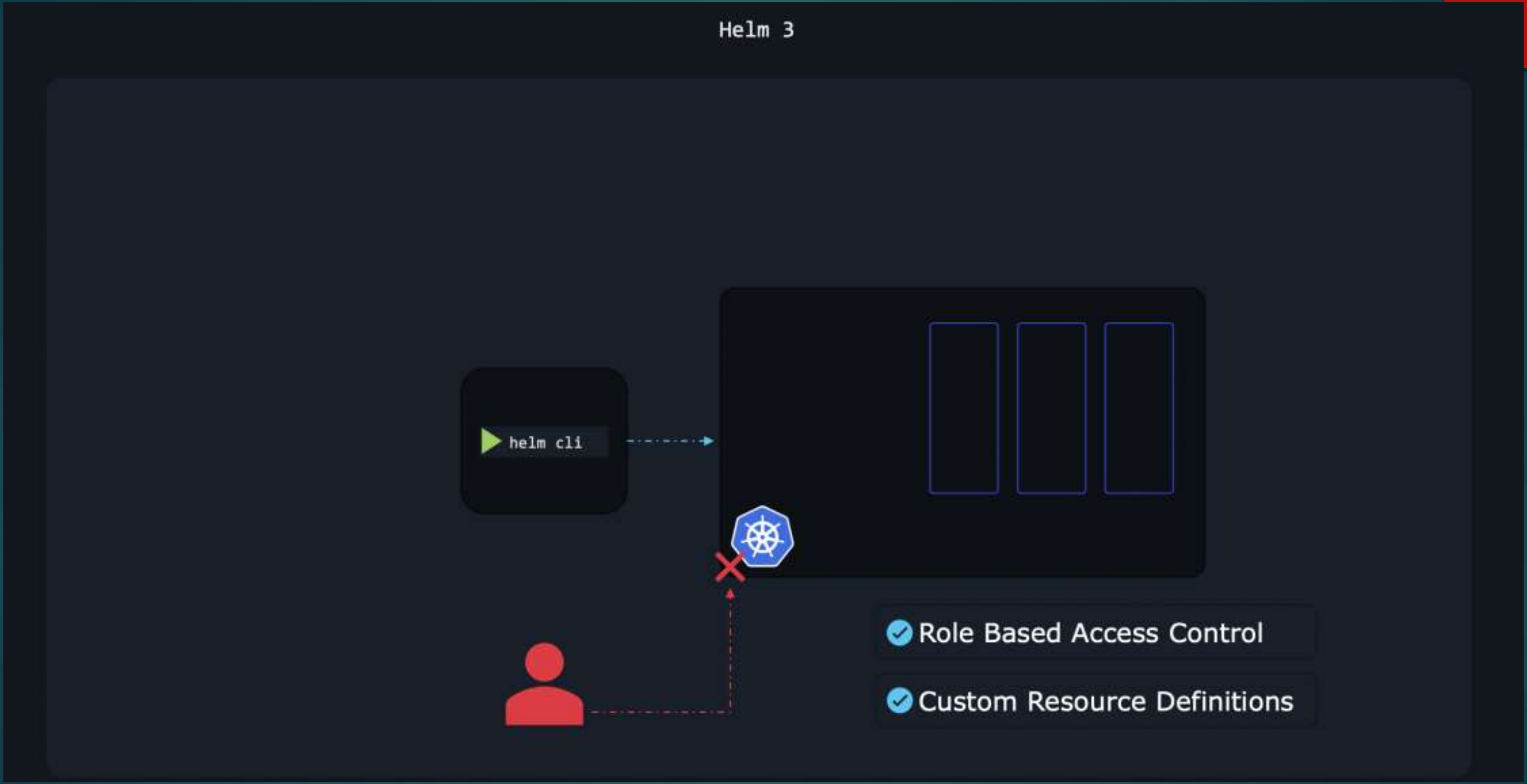
https://helm.sh/docs/intro/install/

HELM 2 VS HELM3



Helm 2

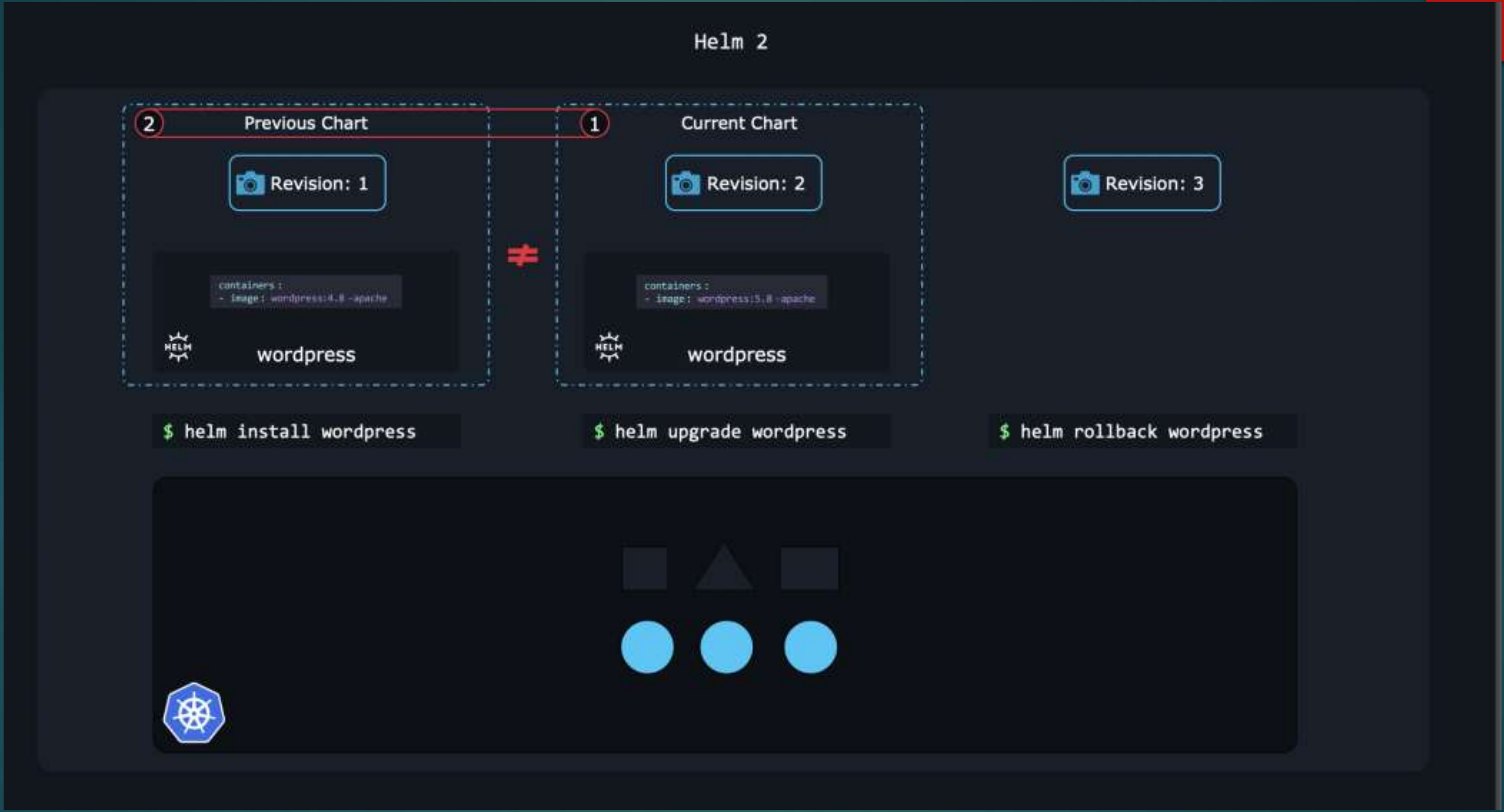
- ✓ Role Based Access Control
- ✓ Custom Resource Definitions



Helm 3

✓ Role Based Access Control

✓ Custom Resource Definitions



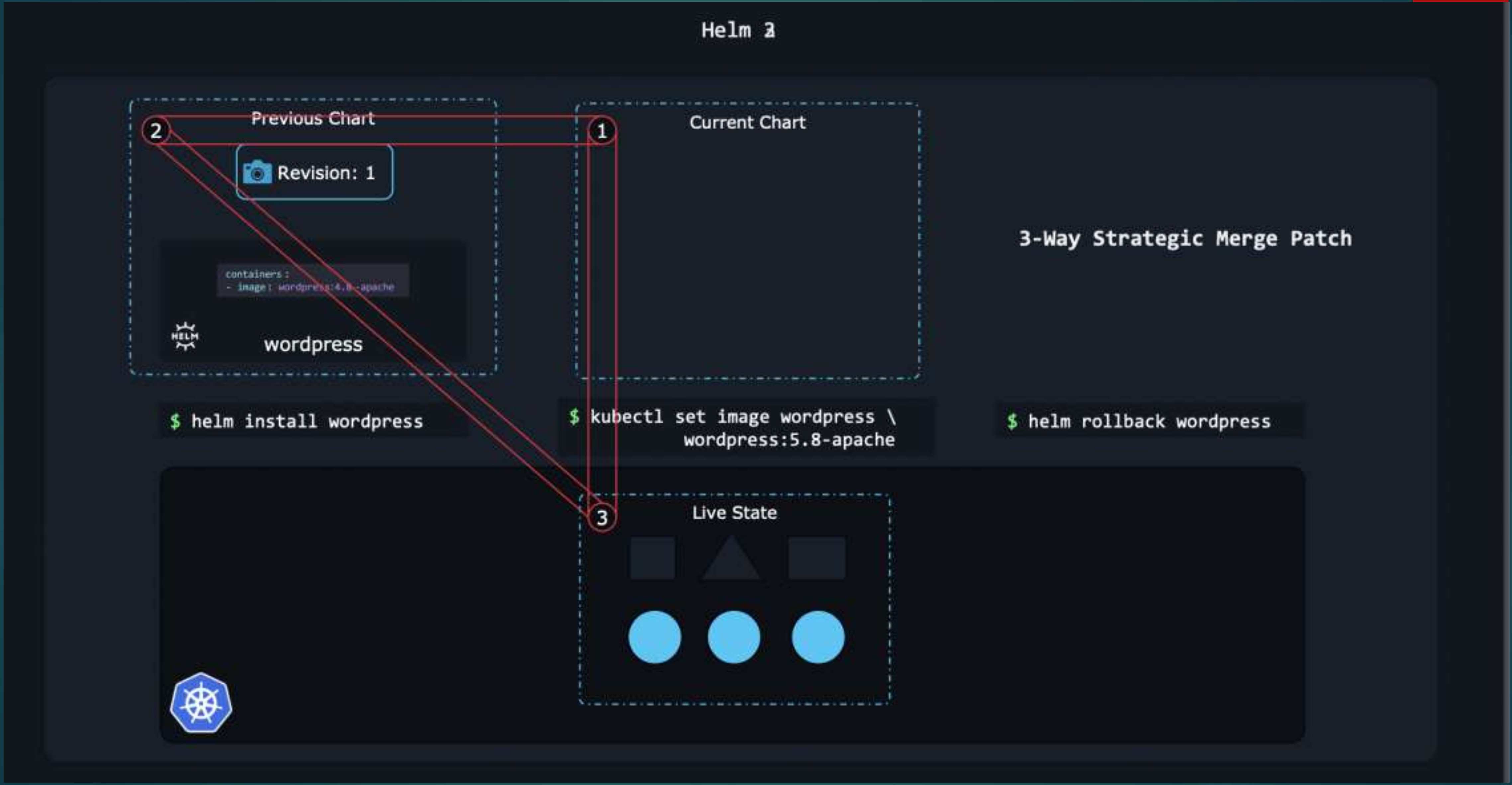
```
containers :
- image : wordpress:4.8 -apache
```

```
containers :
- image : wordpress:5.8 -apache
```

```
$ helm install wordpress
```

```
$ helm upgrade wordpress
```

```
$ helm rollback wordpress
```

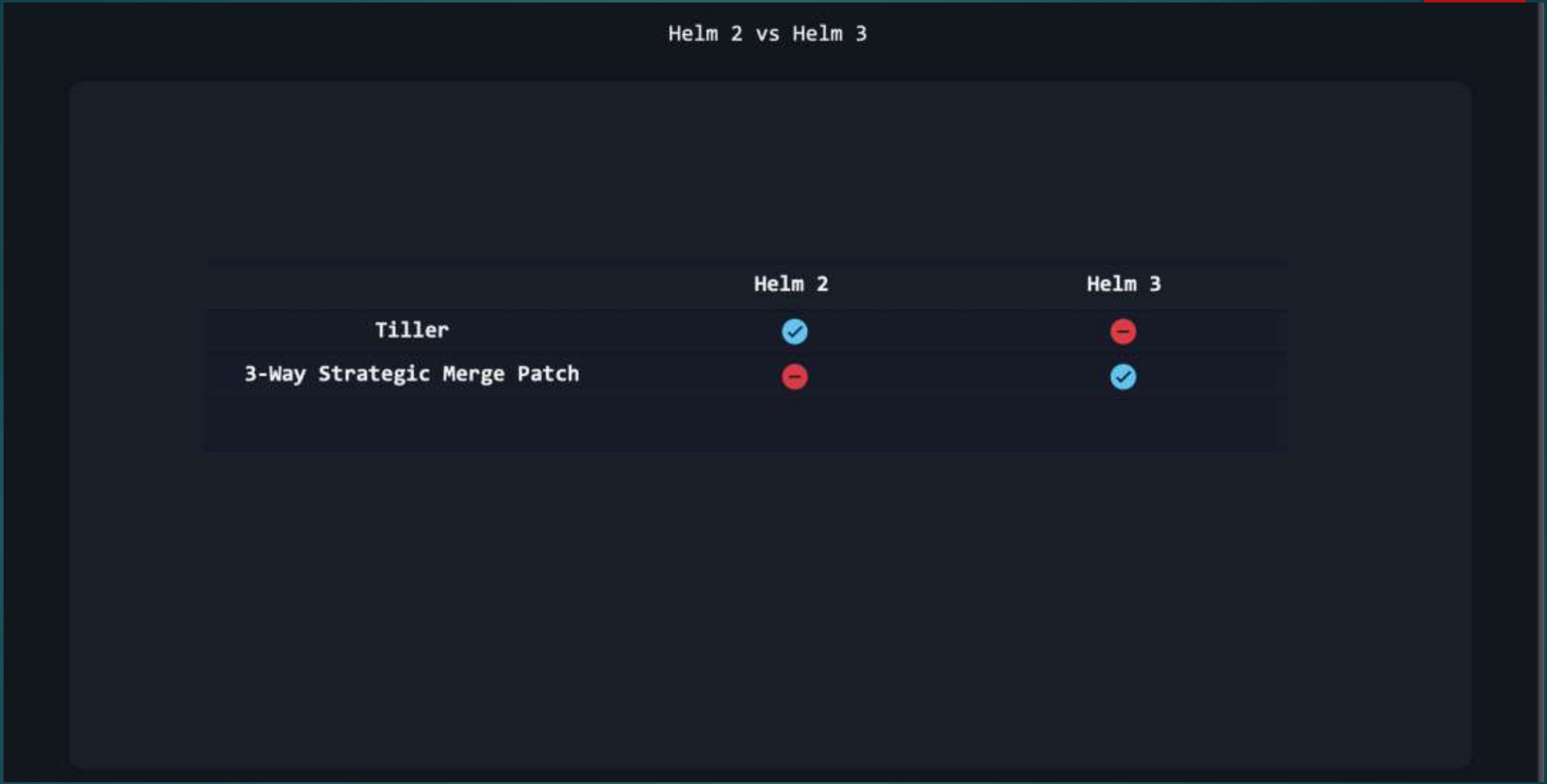


```
containers:
- image: wordpress:4.8-apache
```

```
$ helm install wordpress
```

```
$ kubectl set image wordpress \
          wordpress:5.8-apache
```

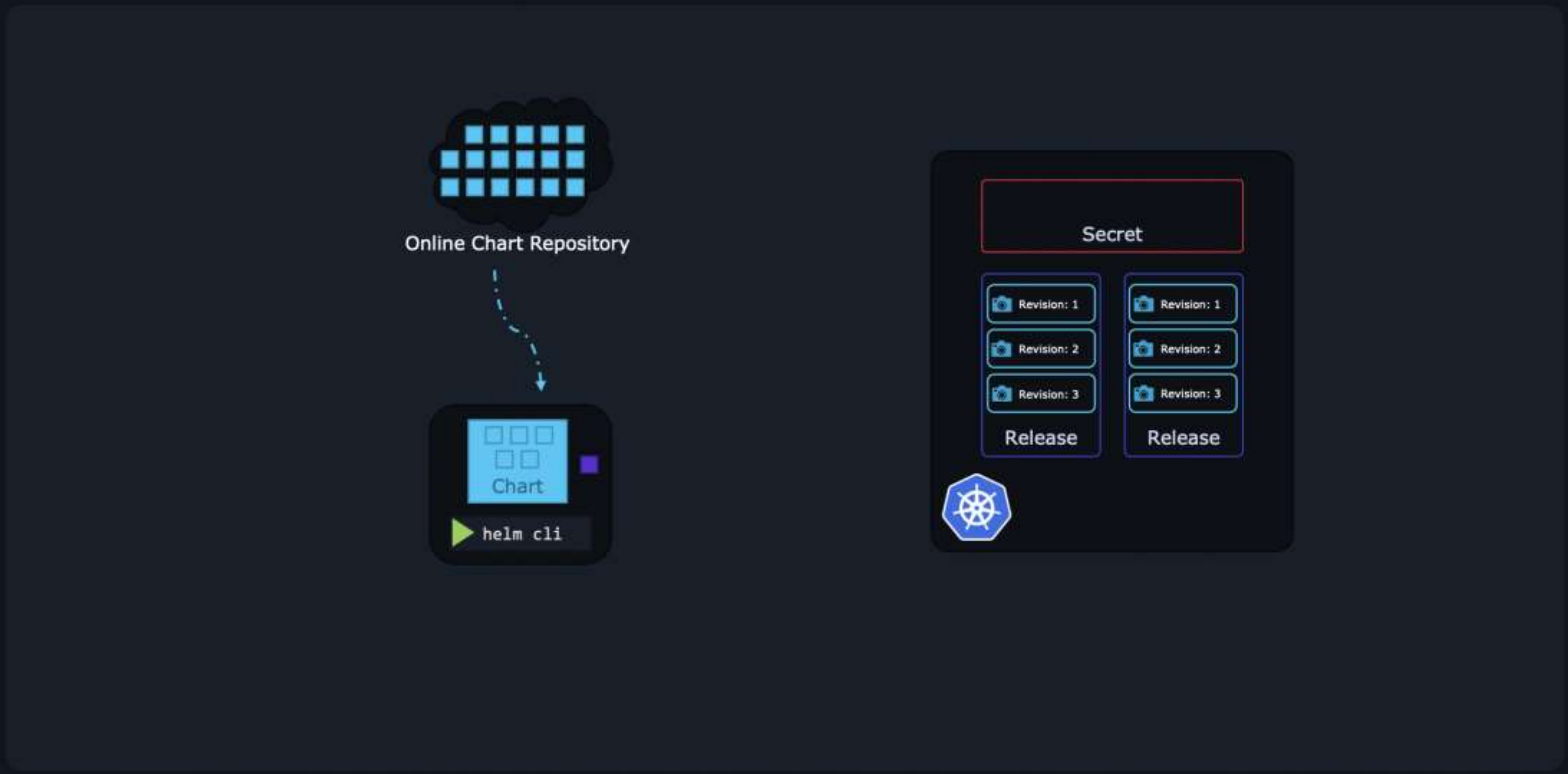
```
$ helm rollback wordpress
```



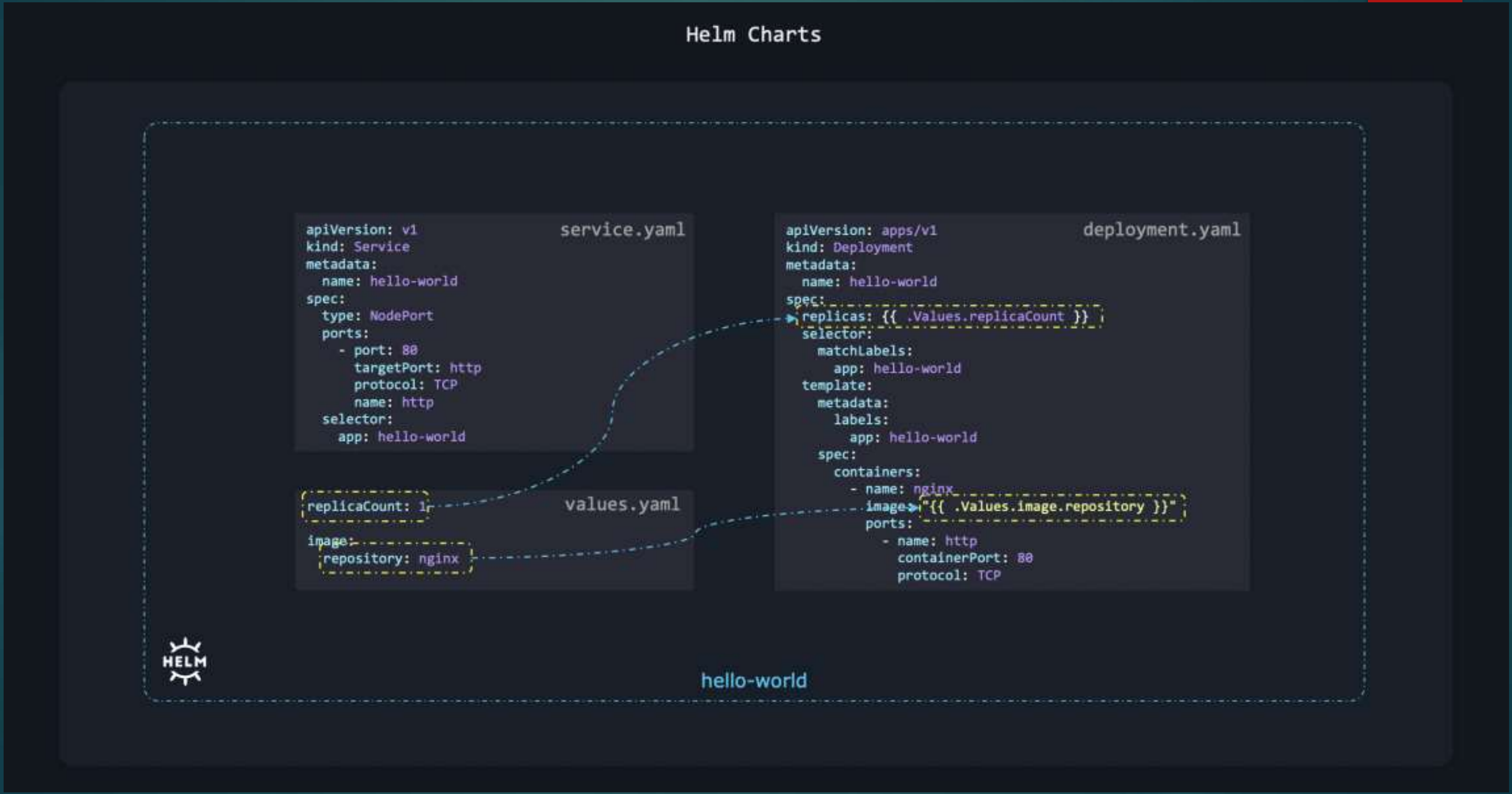
Helm 2 vs Helm 3

	Helm 2	Helm 3
Tiller	✓	⊖
3-Way Strategic Merge Patch	⊖	✓

Helm Components



Helm Charts



service.yaml

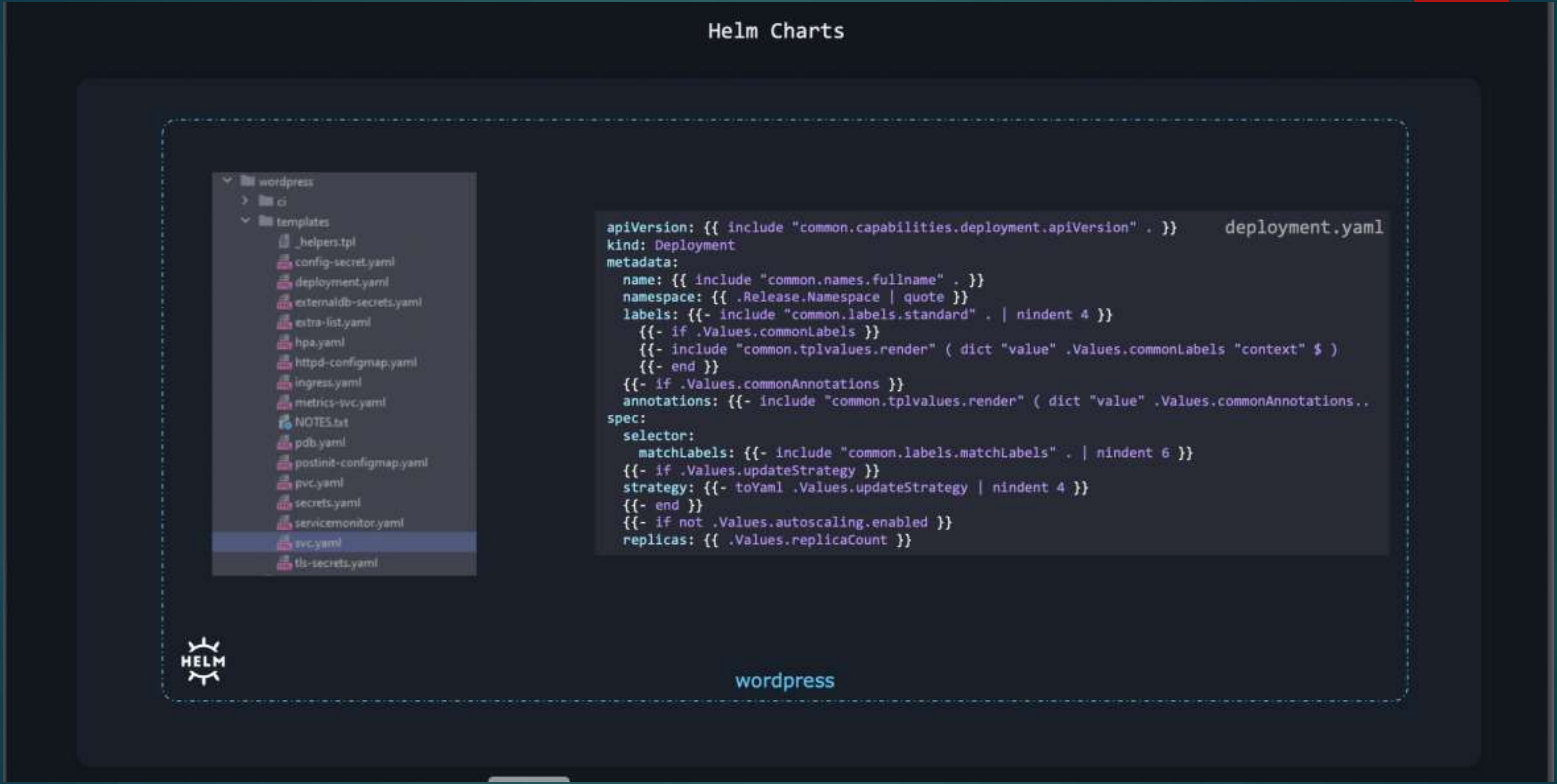
```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-world
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

values.yaml

```yaml
replicaCount: 1
image:
  repository: nginx
```

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: nginx
          image: "{{ .Values.image.repository }}"
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

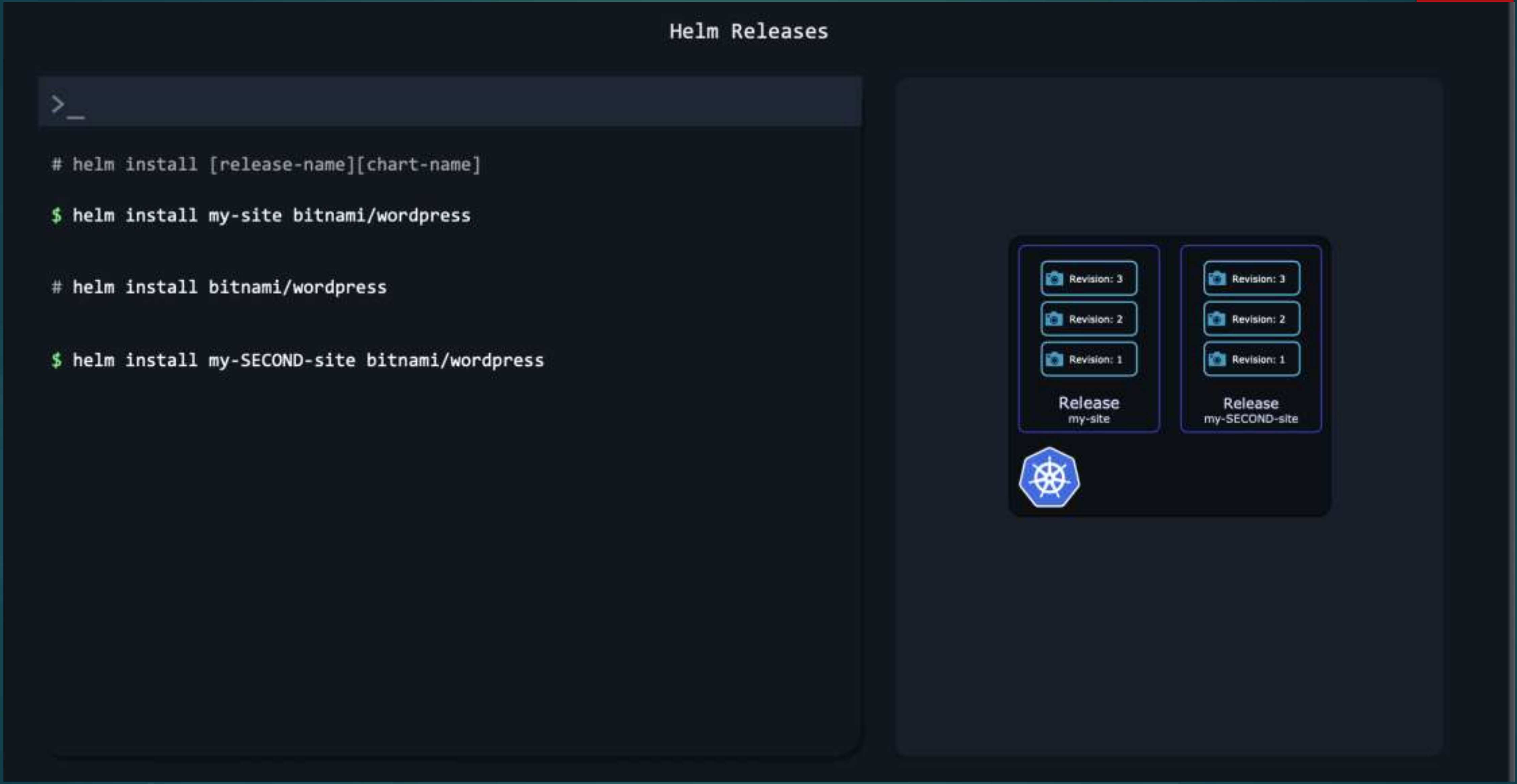


Helm Charts

```
wordpress
  ci
  templates
    _helpers.tpl
    config-secret.yaml
    deployment.yaml
    externaldb-secrets.yaml
    extra-list.yaml
    hpa.yaml
    httpd-configmap.yaml
    ingress.yaml
    metrics-svc.yaml
    NOTES.txt
    pdb.yaml
    postinit-configmap.yaml
    pvc.yaml
    secrets.yaml
    servicemonitor.yaml
    svc.yaml
    tls-secrets.yaml
```

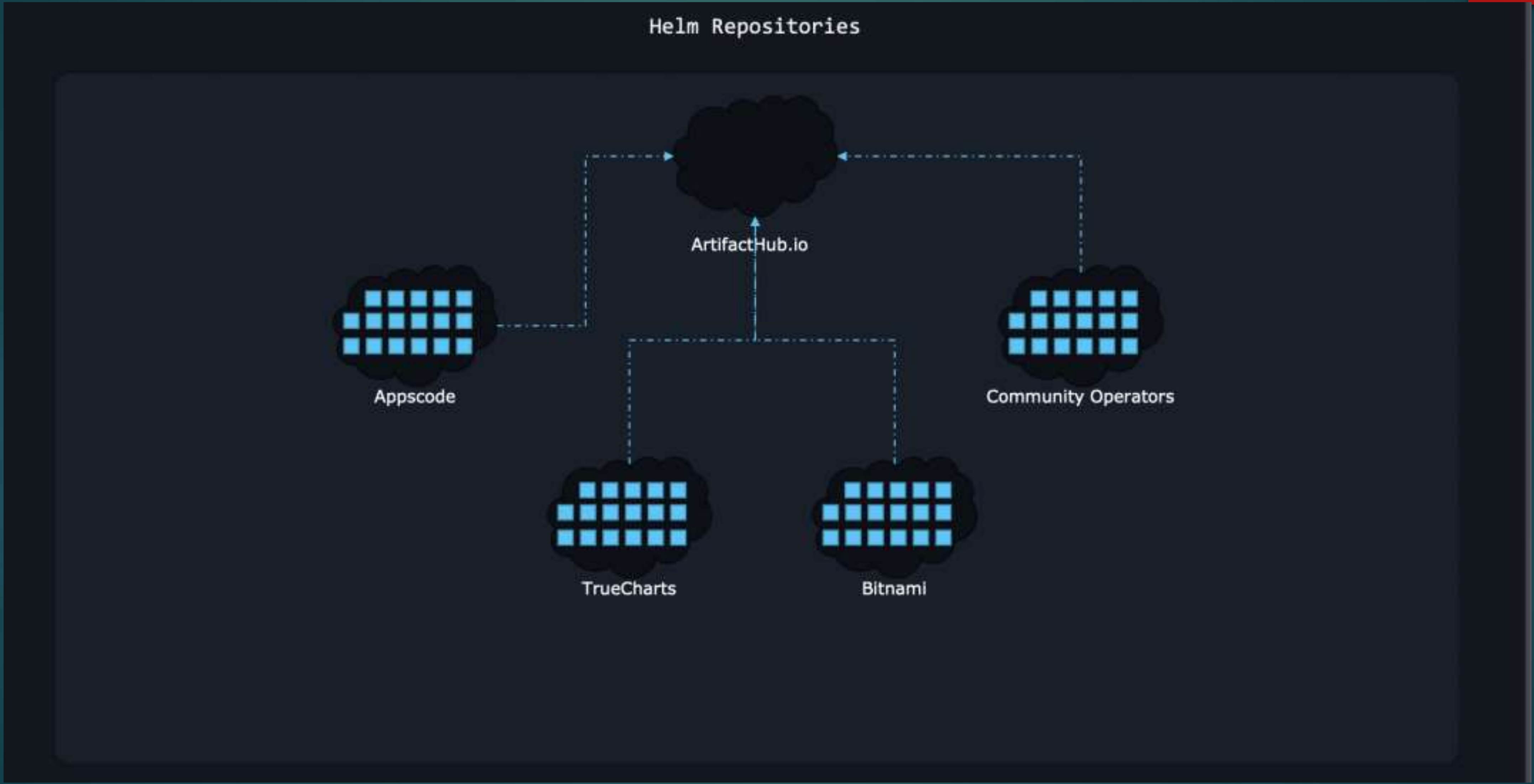
deployment.yaml

```
apiVersion: {{ include "common.capabilities.deployment.apiVersion" . }}
kind: Deployment
metadata:
  name: {{ include "common.names.fullname" . }}
  namespace: {{ .Release.Namespace | quote }}
  labels: {{- include "common.labels.standard" . | nindent 4 }}
    {{- if .Values.commonLabels }}
    {{- include "common.tplvalues.render" ( dict "value" .Values.commonLabels "context" $ )
    {{- end }}
  {{- if .Values.commonAnnotations }}
  annotations: {{- include "common.tplvalues.render" ( dict "value" .Values.commonAnnotations..
spec:
  selector:
    matchLabels: {{- include "common.labels.matchLabels" . | nindent 6 }}
  {{- if .Values.updateStrategy }}
  strategy: {{- toYaml .Values.updateStrategy | nindent 4 }}
  {{- end }}
  {{- if not .Values.autoscaling.enabled }}
  replicas: {{ .Values.replicaCount }}
```

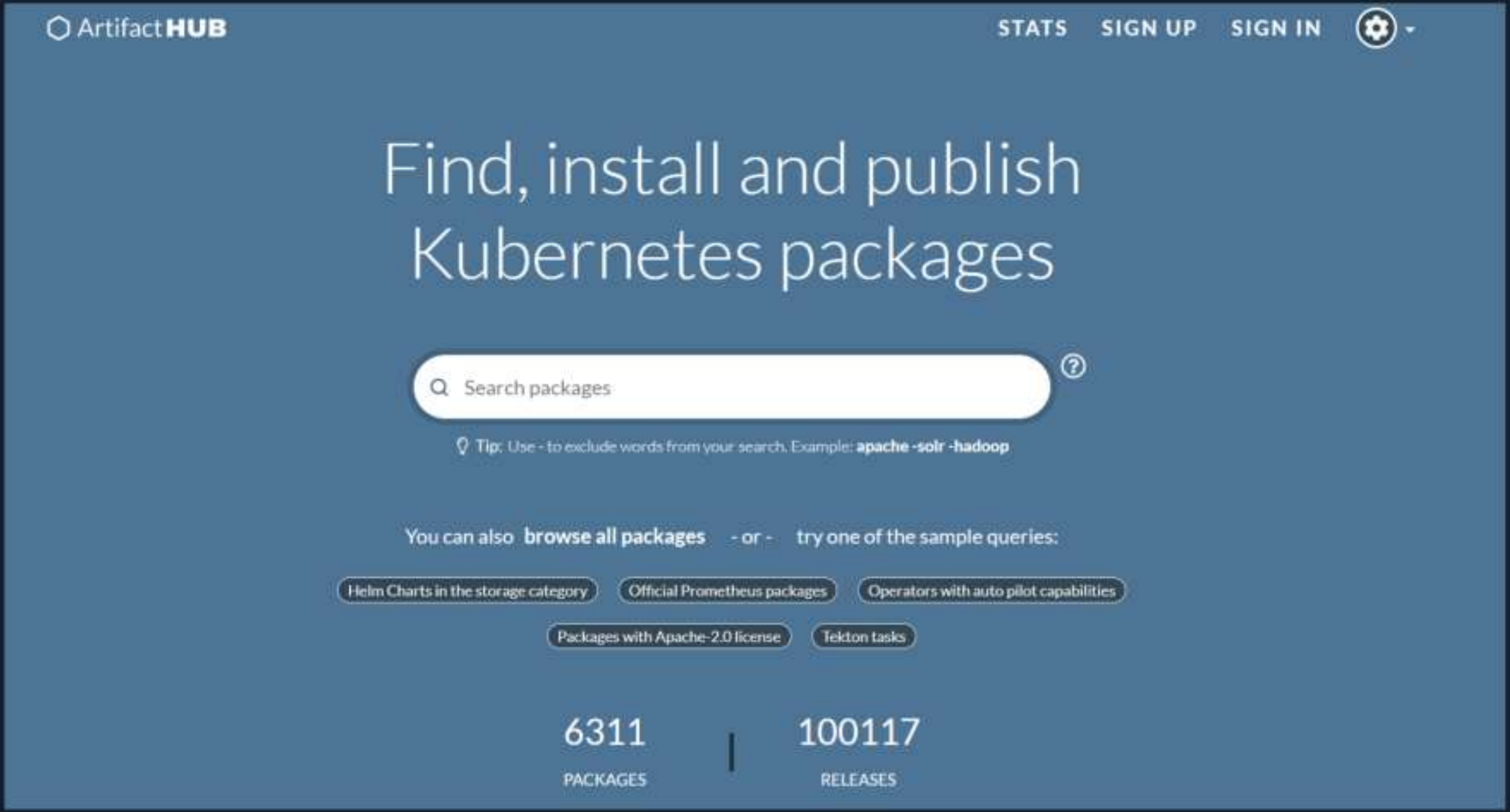


Helm Releases

```
>_

# helm install [release-name][chart-name]

$ helm install my-site bitnami/wordpress


# helm install bitnami/wordpress


$ helm install my-SECOND-site bitnami/wordpress
```


Helm Repositories
ArtifactHub.io
Appscode
Community Operators
TrueCharts
Bitnami

Helm Repositories


ArtifactHUB
STATS
SIGN UP
SIGN IN
Search packages

Find, install and publish Kubernetes packages

Tip: Use - to exclude words from your search. Example: **apache -solr -hadoop**

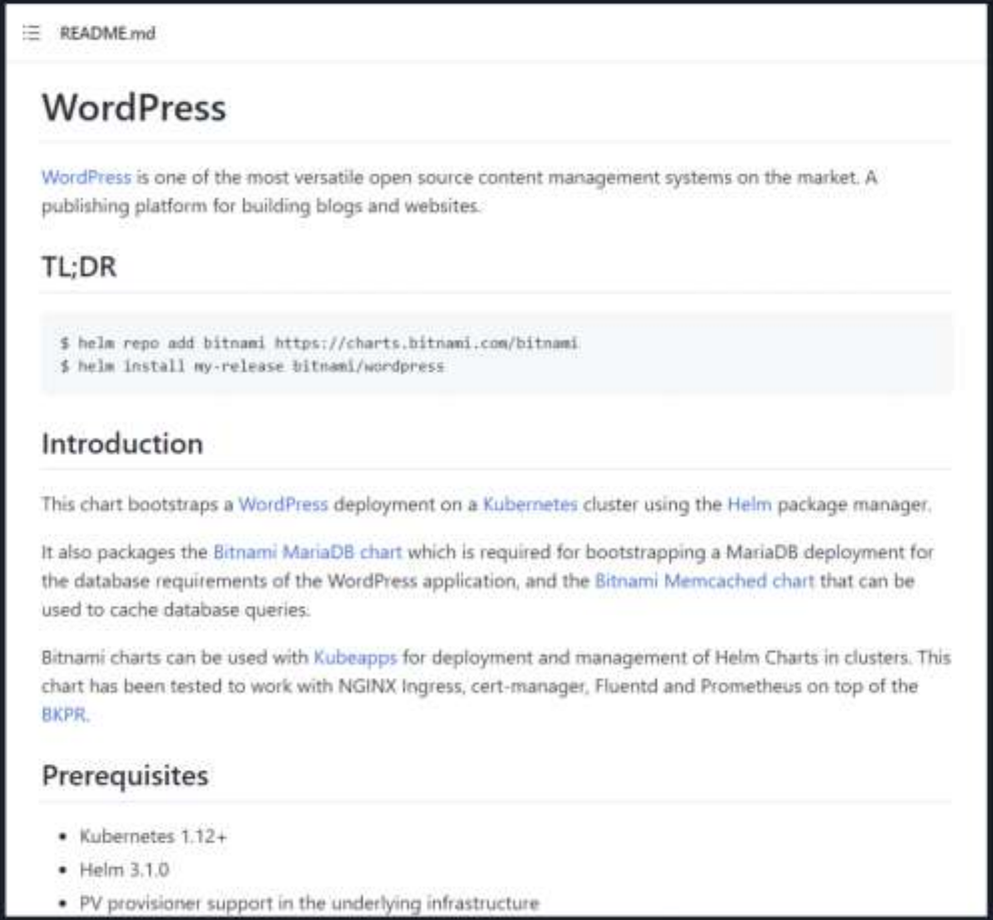
You can also **browse all packages** - or - try one of the sample queries:

Helm Charts in the storage category
Official Prometheus packages
Operators with auto pilot capabilities
Packages with Apache-2.0 license
Tekton tasks

6311 PACKAGES | 100117 RELEASES

Helm Chart Structure

```
hello-world-chart
    templates      # Templates directory
    values.yaml    # Configurable values
    Chart.yaml     # Chart information
    LICENSE        # Chart License
    README.md      # Readme file
    charts         # Dependency Charts
```



README.md

WordPress

WordPress is one of the most versatile open source content management systems on the market. A publishing platform for building blogs and websites.

TL;DR

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm install my-release bitnami/wordpress
```

Introduction

This chart bootstraps a WordPress deployment on a Kubernetes cluster using the Helm package manager.

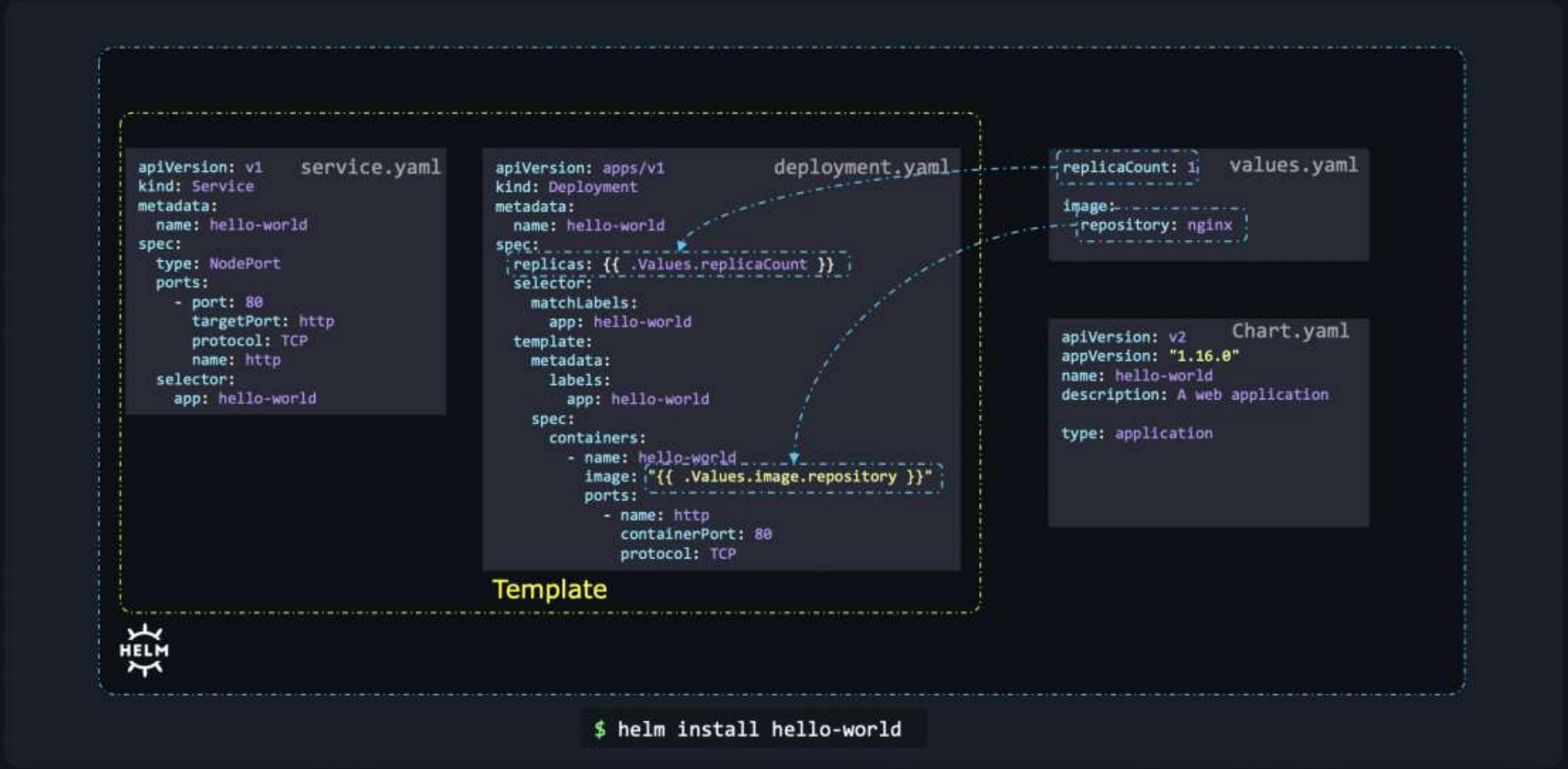
It also packages the Bitnami MariaDB chart which is required for bootstrapping a MariaDB deployment for the database requirements of the WordPress application, and the Bitnami Memcached chart that can be used to cache database queries.

Bitnami charts can be used with Kubeapps for deployment and management of Helm Charts in clusters. This chart has been tested to work with NGINX Ingress, cert-manager, Fluentd and Prometheus on top of the BKPR.

Prerequisites

- Kubernetes 1.12+
- Helm 3.1.0
- PV provisioner support in the underlying infrastructure

Helm Charts



service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-world
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: "{{ .Values.image.repository }}"
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

values.yaml

```yaml
replicaCount: 1
image:
  repository: nginx
```

Chart.yaml

```yaml
apiVersion: v2
appVersion: "1.16.0"
name: hello-world
description: A web application

type: application
```

```
$ helm install hello-world
```

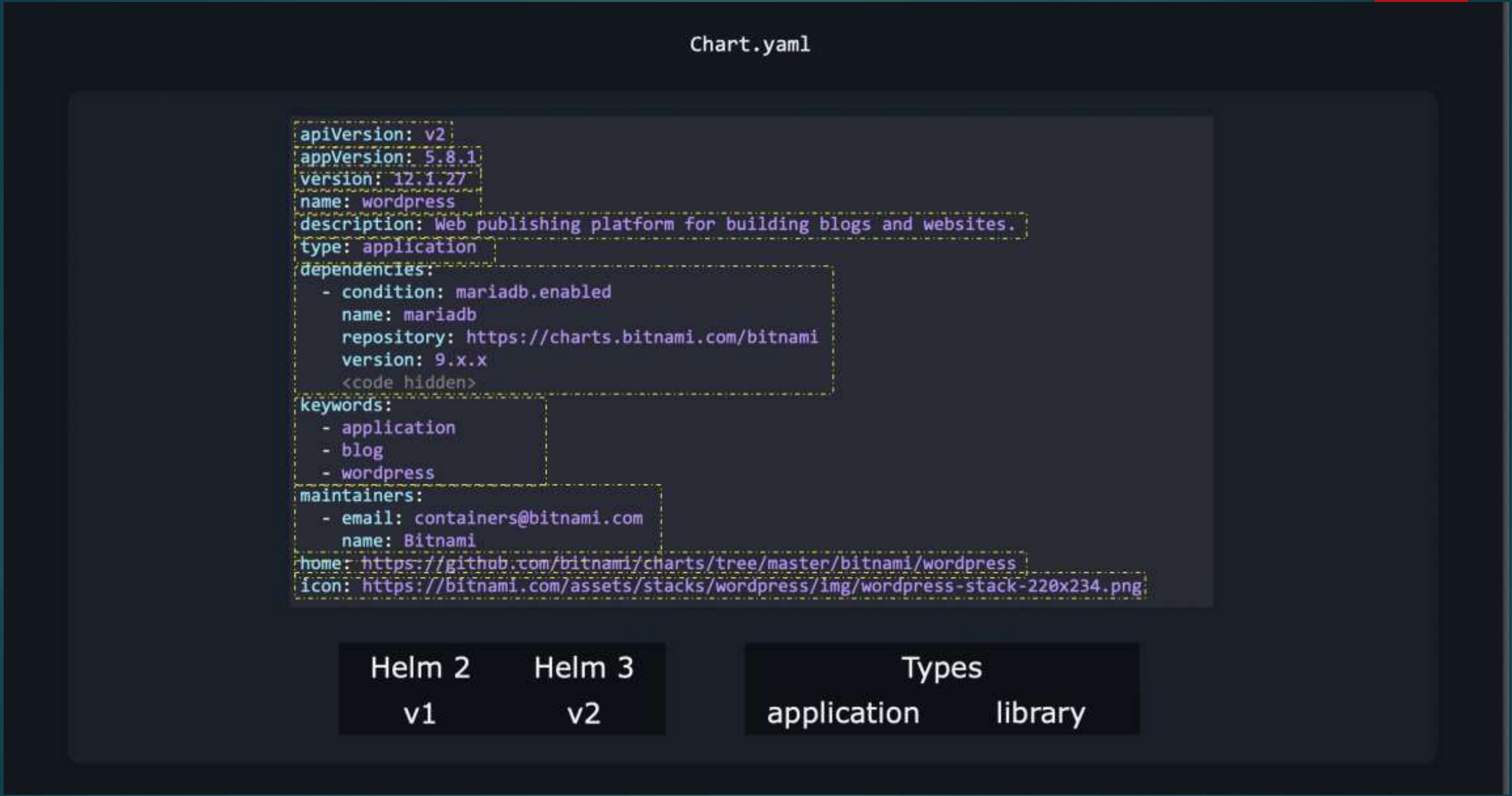


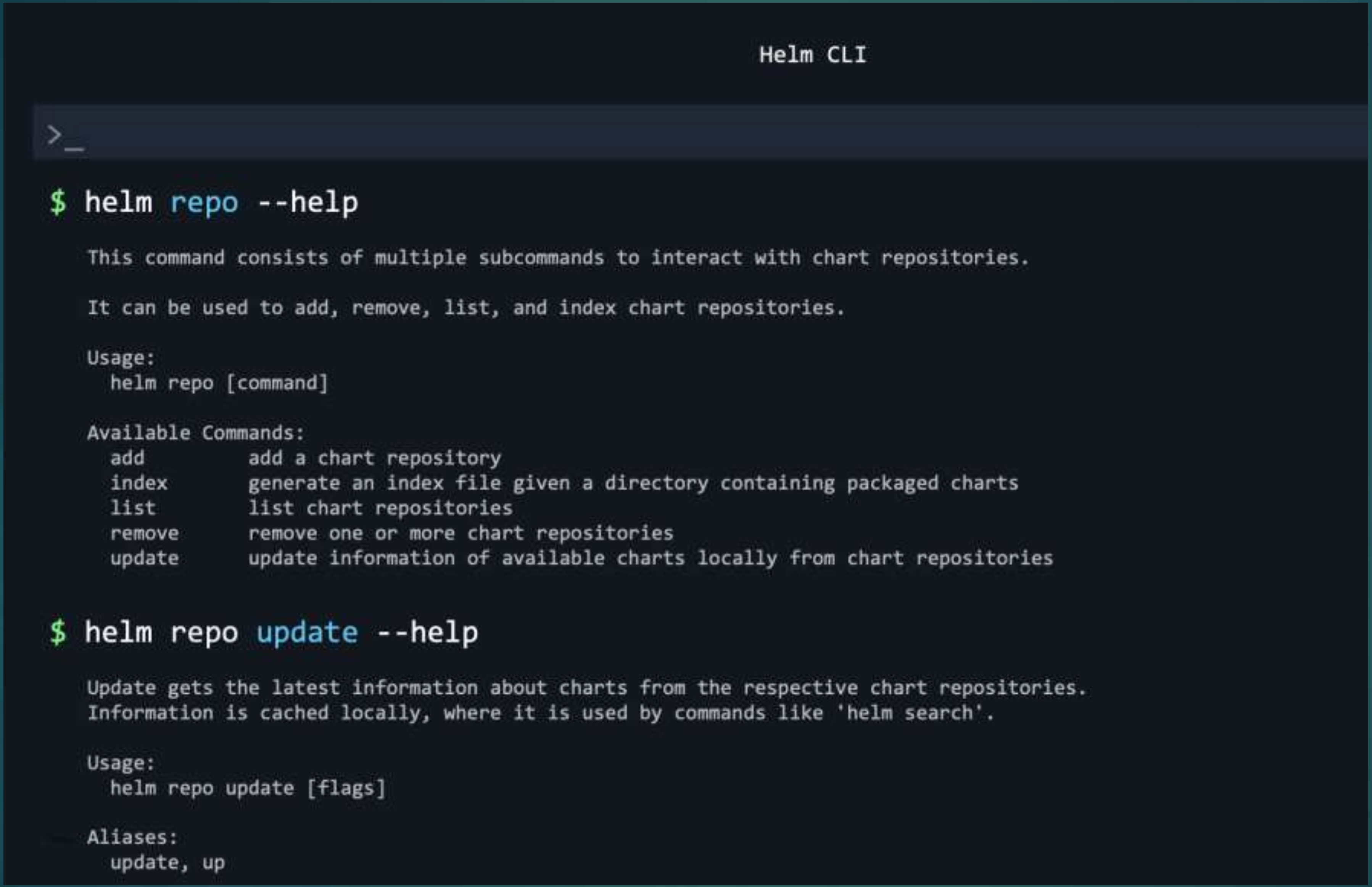
Chart.yaml

```yaml
apiVersion: v2
appVersion: 5.8.1
version: 12.1.27
name: wordpress
description: Web publishing platform for building blogs and websites.
type: application
dependencies:
  - condition: mariadb.enabled
    name: mariadb
    repository: https://charts.bitnami.com/bitnami
    version: 9.x.x
    <code hidden>
keywords:
  - application
  - blog
  - wordpress
maintainers:
  - email: containers@bitnami.com
    name: Bitnami
home: https://github.com/bitnami/charts/tree/master/bitnami/wordpress
icon: https://bitnami.com/assets/stacks/wordpress/img/wordpress-stack-220x234.png
```

Helm 2	Helm 3
v1	v2

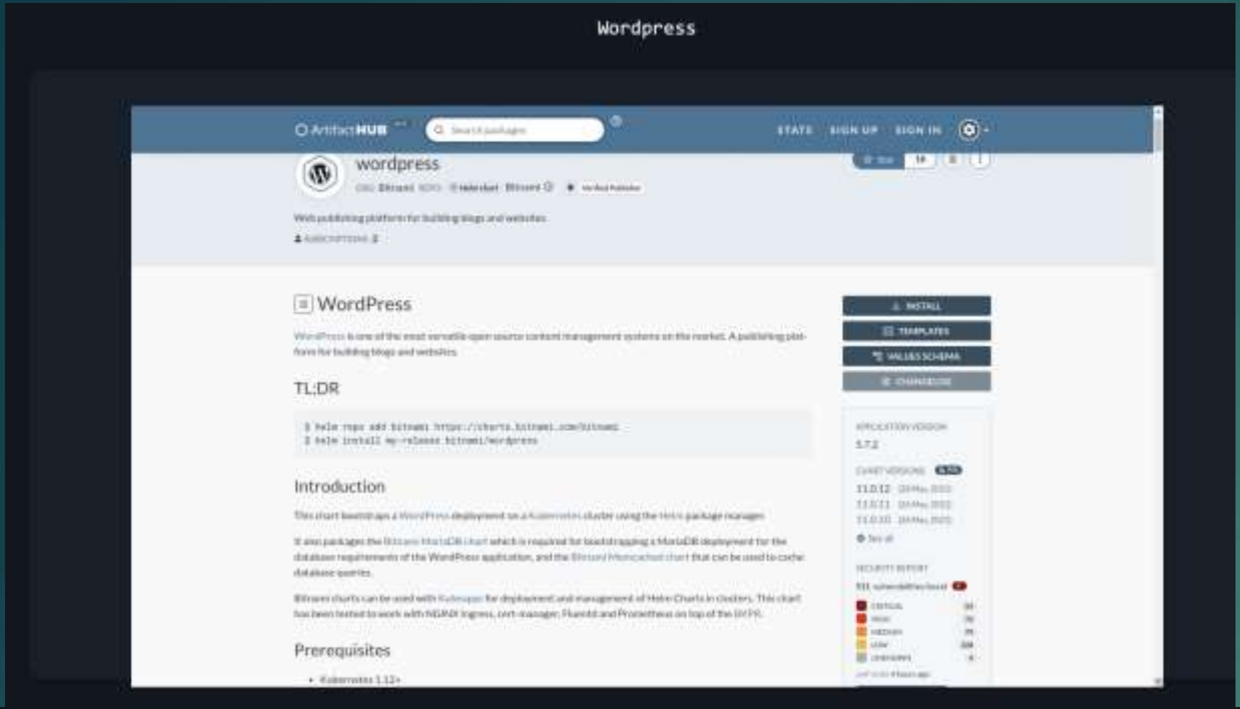
Types	
application	library

Working With HELM



Helm CLI

```
>_

$ helm repo --help

  This command consists of multiple subcommands to interact with chart repositories.

  It can be used to add, remove, list, and index chart repositories.

  Usage:
    helm repo [command]

  Available Commands:
    add         add a chart repository
    index       generate an index file given a directory containing packaged charts
    list        list chart repositories
    remove      remove one or more chart repositories
    update      update information of available charts locally from chart repositories

$ helm repo update --help

  Update gets the latest information about charts from the respective chart repositories.
  Information is cached locally, where it is used by commands like 'helm search'.

  Usage:
    helm repo update [flags]

  Aliases:
    update, up
```



Web publishing platform for building blogs and websites.

WordPress

WordPress is one of the most versatile open source content management systems on the market. A publishing platform for building blogs and websites.

TL;DR

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm install my-release bitnami/wordpress
```

Introduction

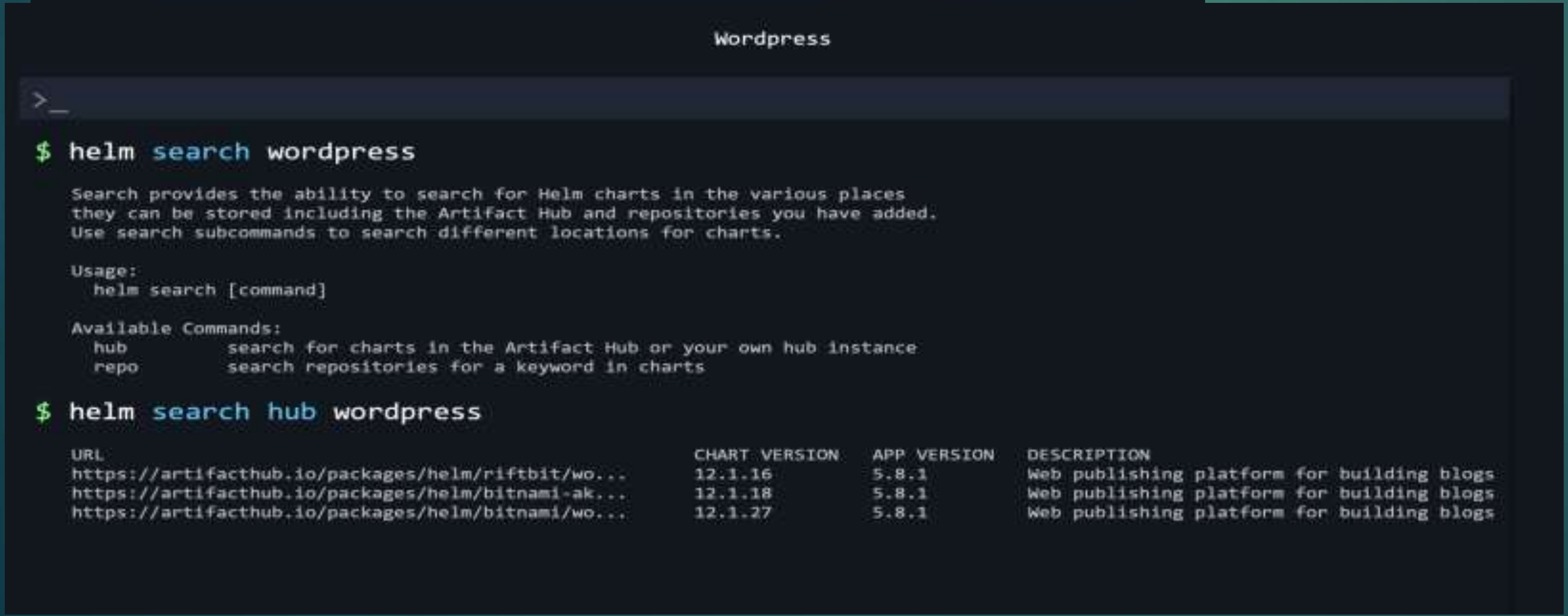
This chart bootstraps a WordPress deployment on a Kubernetes cluster using the Helm package manager.

It also packages the Bitnami MariaDB chart which is required for bootstrapping a MariaDB deployment for the database requirements of the WordPress application, and the Bitnami Memcached chart that can be used to cache database queries.

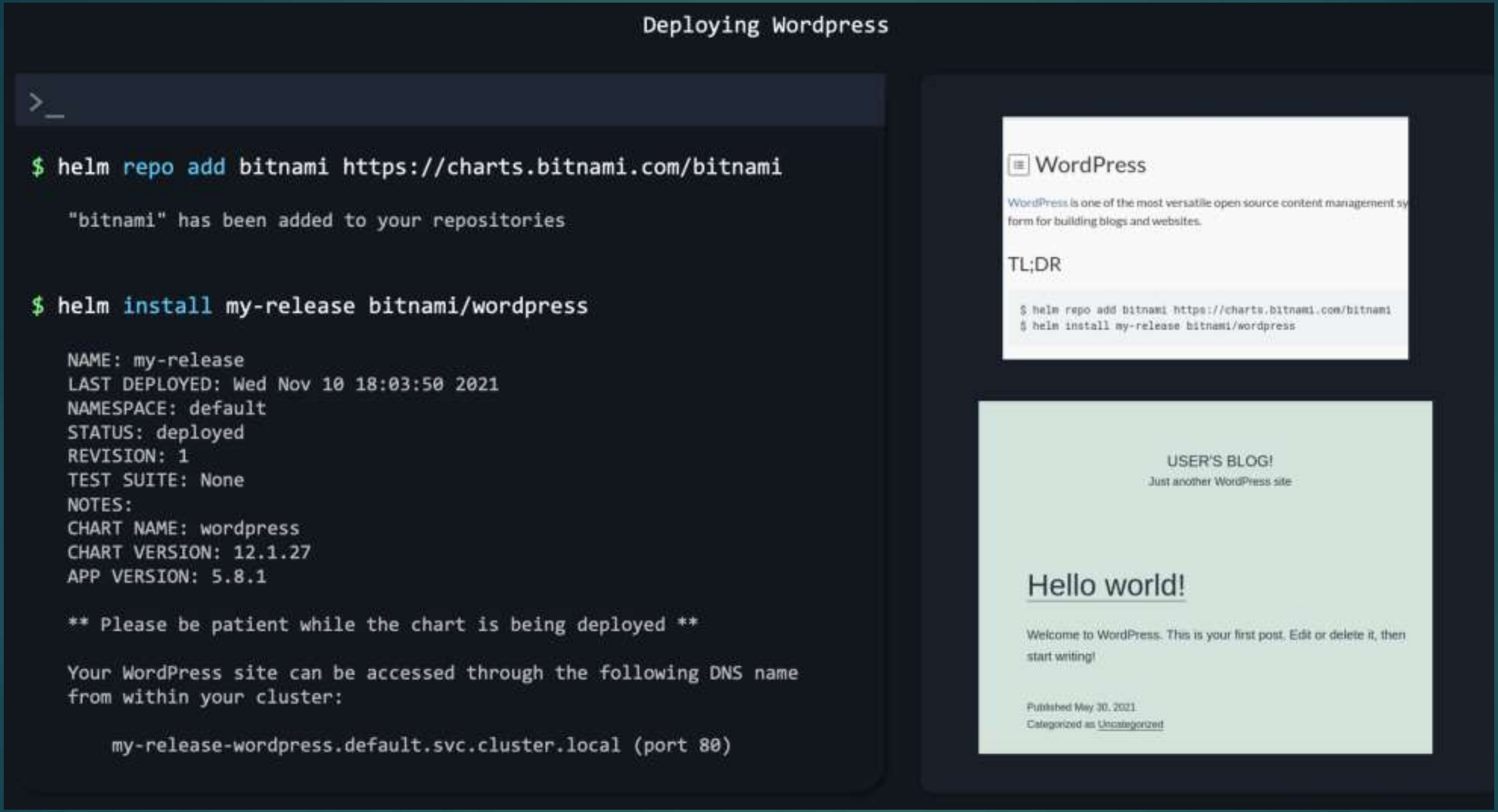
Bitnami charts can be used with Kubeapps for deployment and management of Helm Charts in clusters. This chart has been tested to work with NGINX Ingress, cert-manager, fluentd and Prometheus on top of the BKPR.

Prerequisites

- Kubernetes 1.12+



```
$ helm search wordpress

  Search provides the ability to search for Helm charts in the various places
  they can be stored including the Artifact Hub and repositories you have added.
  Use search subcommands to search different locations for charts.

  Usage:
    helm search [command]

  Available Commands:
    hub         search for charts in the Artifact Hub or your own hub instance
    repo        search repositories for a keyword in charts

$ helm search hub wordpress

  URL                                                   CHART VERSION   APP VERSION   DESCRIPTION
  https://artifacthub.io/packages/helm/riftbit/wo...    12.1.16         5.8.1         Web publishing platform for building blogs
  https://artifacthub.io/packages/helm/bitnami-ak...    12.1.18         5.8.1         Web publishing platform for building blogs
  https://artifacthub.io/packages/helm/bitnami/wo...    12.1.27         5.8.1         Web publishing platform for building blogs
```



Deploying Wordpress

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami

   "bitnami" has been added to your repositories


$ helm install my-release bitnami/wordpress

   NAME: my-release
   LAST DEPLOYED: Wed Nov 10 18:03:50 2021
   NAMESPACE: default
   STATUS: deployed
   REVISION: 1
   TEST SUITE: None
   NOTES:
   CHART NAME: wordpress
   CHART VERSION: 12.1.27
   APP VERSION: 5.8.1

   ** Please be patient while the chart is being deployed **

   Your WordPress site can be accessed through the following DNS name
   from within your cluster:

       my-release-wordpress.default.svc.cluster.local (port 80)
```

WordPress

WordPress is one of the most versatile open source content management sy
form for building blogs and websites.

TL;DR

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm install my-release bitnami/wordpress
```

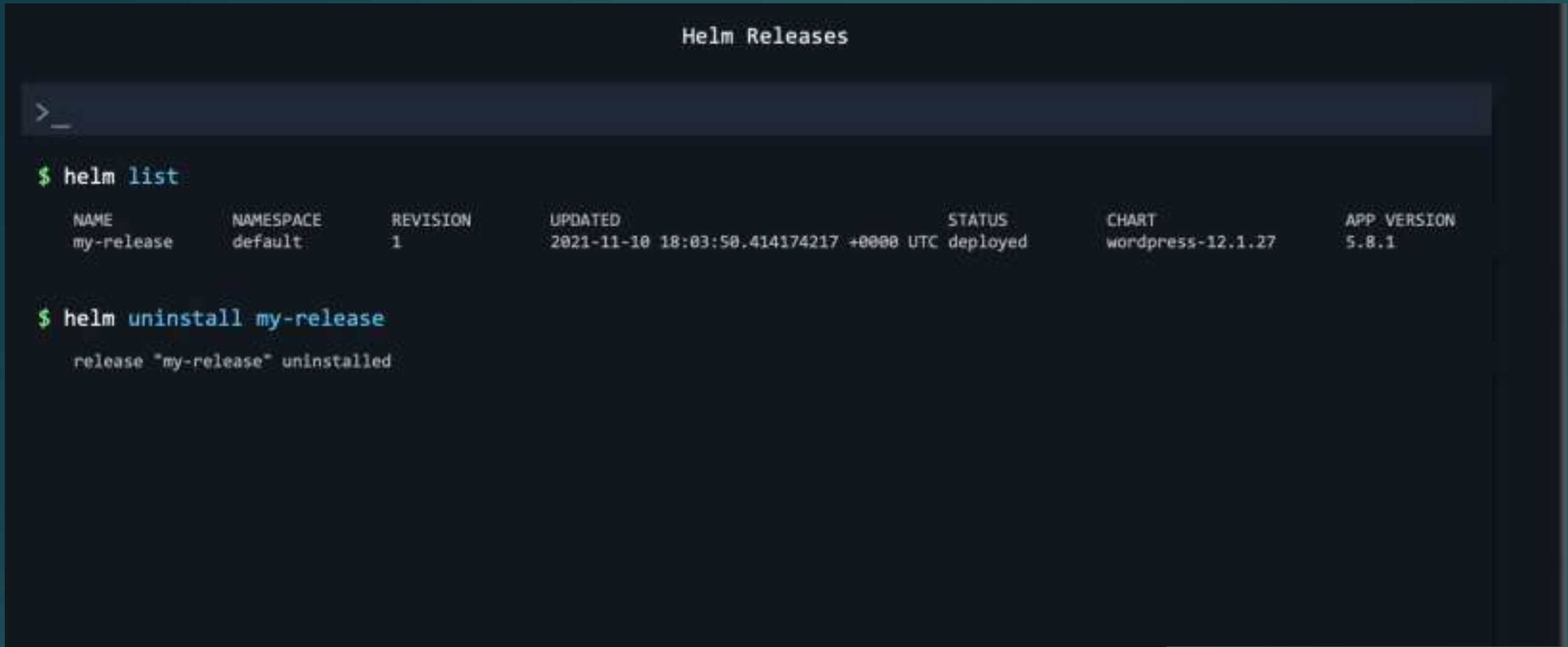
USER'S BLOG!

Just another WordPress site

Hello world!

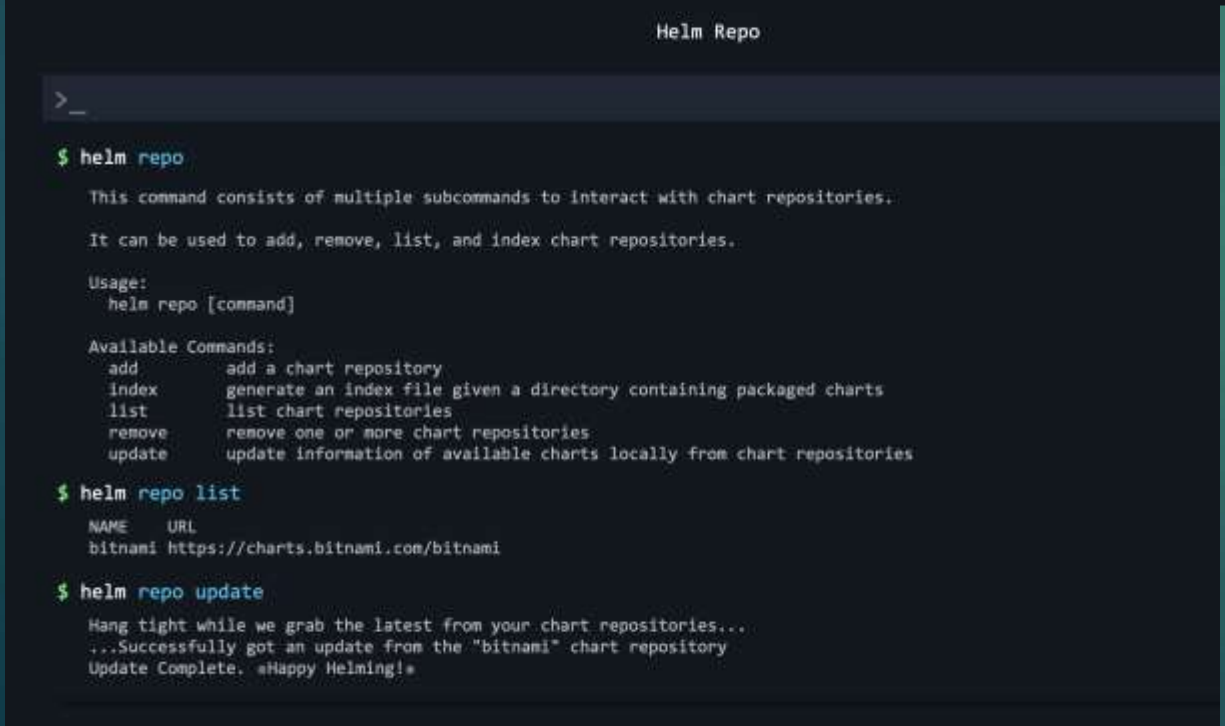
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Published May 30, 2021
Categorized as Uncategorized



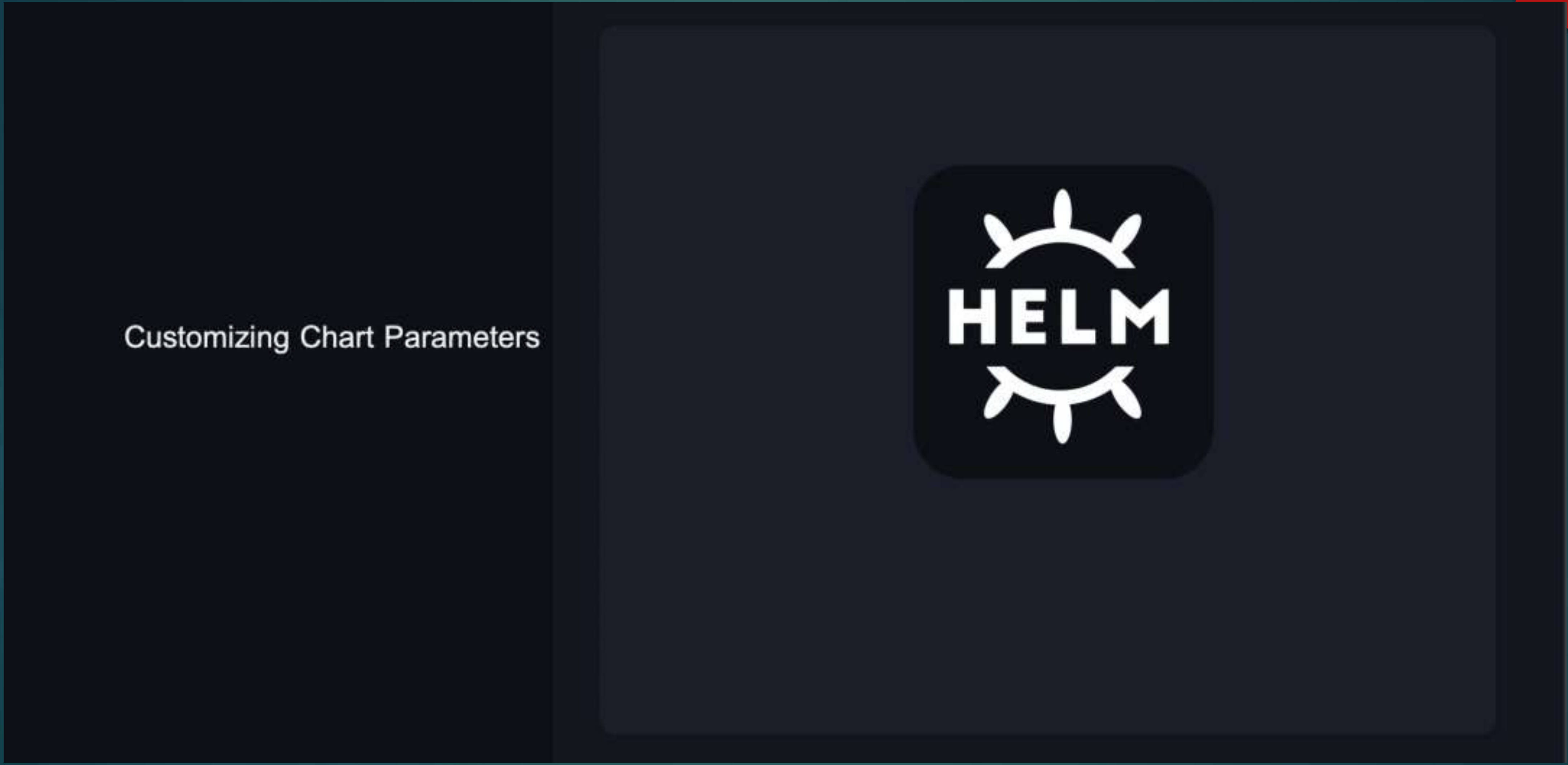
Helm Releases

```
>_

$ helm list

  NAME          NAMESPACE     REVISION      UPDATED                                   STATUS          CHART                  APP VERSION
  my-release    default       1             2021-11-10 18:03:50.414174217 +0000 UTC deployed        wordpress-12.1.27      5.8.1

$ helm uninstall my-release

  release "my-release" uninstalled
```

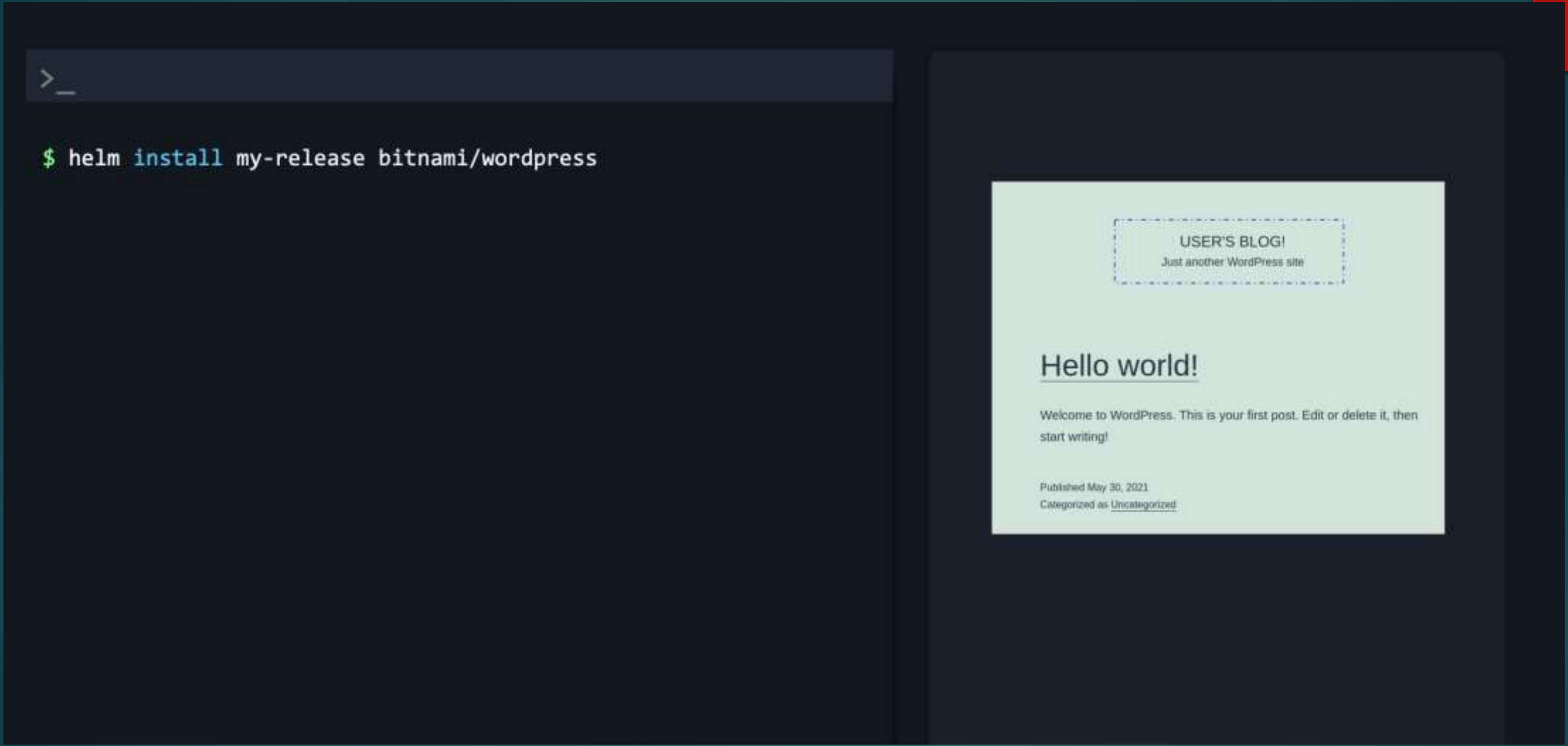


Helm Repo

```
>_

$ helm repo
   This command consists of multiple subcommands to interact with chart repositories.

   It can be used to add, remove, list, and index chart repositories.

   Usage:
     helm repo [command]

   Available Commands:
     add         add a chart repository
     index       generate an index file given a directory containing packaged charts
     list        list chart repositories
     remove      remove one or more chart repositories
     update      update information of available charts locally from chart repositories
$ helm repo list
   NAME    URL
   bitnami https://charts.bitnami.com/bitnami
$ helm repo update
   Hang tight while we grab the latest from your chart repositories...
   ...Successfully got an update from the "bitnami" chart repository
   Update Complete. ⎈Happy Helming!⎈
```



Customizing Chart Parameters



```
$ helm install my-release bitnami/wordpress
```

USER'S BLOG!

Just another WordPress site

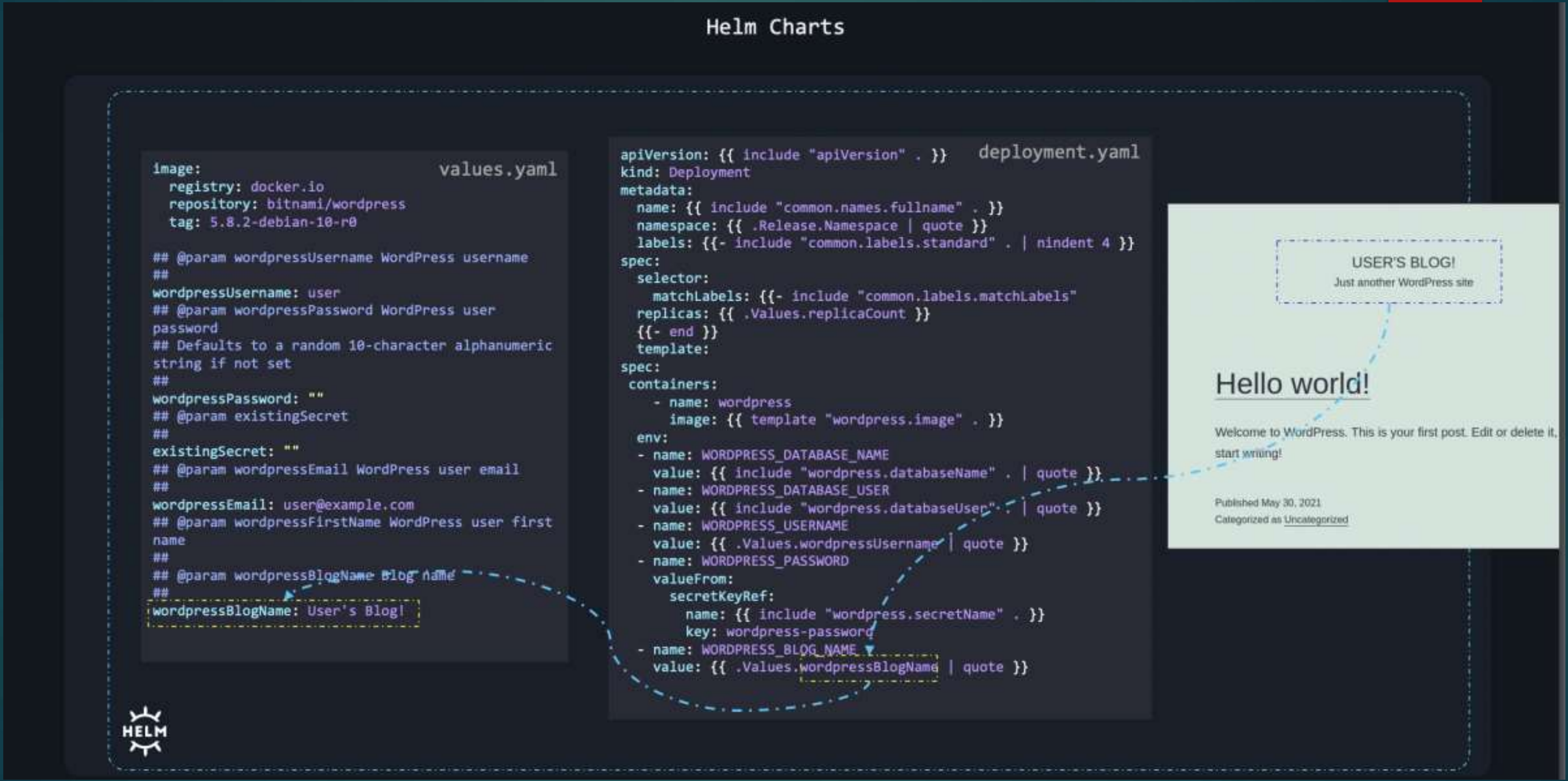
Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Published May 30, 2021

Categorized as Uncategorized

Helm Charts



values.yaml

```
image:
  registry: docker.io
  repository: bitnami/wordpress
  tag: 5.8.2-debian-10-r0

## @param wordpressUsername WordPress username
##
wordpressUsername: user
## @param wordpressPassword WordPress user
password
## Defaults to a random 10-character alphanumeric
string if not set
##
wordpressPassword: ""
## @param existingSecret
##
existingSecret: ""
## @param wordpressEmail WordPress user email
##
wordpressEmail: user@example.com
## @param wordpressFirstName WordPress user first
name
##
## @param wordpressBlogName Blog name
##
wordpressBlogName: User's Blog!
```

deployment.yaml

```
apiVersion: {{ include "apiVersion" . }}
kind: Deployment
metadata:
  name: {{ include "common.names.fullname" . }}
  namespace: {{ .Release.Namespace | quote }}
  labels: {{- include "common.labels.standard" . | nindent 4 }}
spec:
  selector:
    matchLabels: {{- include "common.labels.matchLabels"
  replicas: {{ .Values.replicaCount }}
  {{- end }}
  template:
spec:
 containers:
    - name: wordpress
      image: {{ template "wordpress.image" . }}
  env:
  - name: WORDPRESS_DATABASE_NAME
    value: {{ include "wordpress.databaseName" . | quote }}
  - name: WORDPRESS_DATABASE_USER
    value: {{ include "wordpress.databaseUser" . | quote }}
  - name: WORDPRESS_USERNAME
    value: {{ .Values.wordpressUsername | quote }}
  - name: WORDPRESS_PASSWORD
    valueFrom:
      secretKeyRef:
        name: {{ include "wordpress.secretName" . }}
        key: wordpress-password
  - name: WORDPRESS_BLOG_NAME
    value: {{ .Values.wordpressBlogName | quote }}
```

USER'S BLOG!

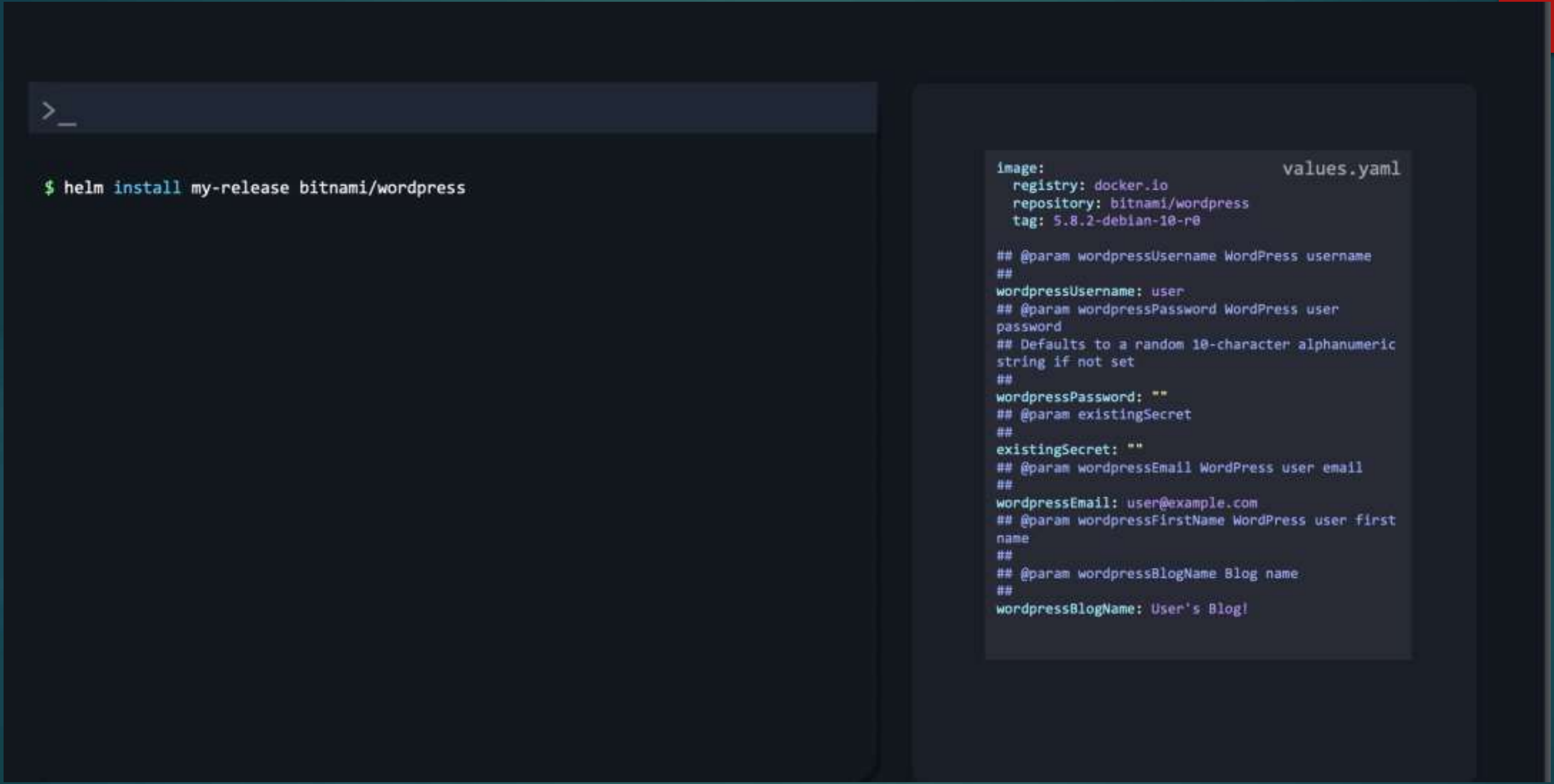
Just another WordPress site

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, start writing!

Published May 30, 2021

Categorized as Uncategorized

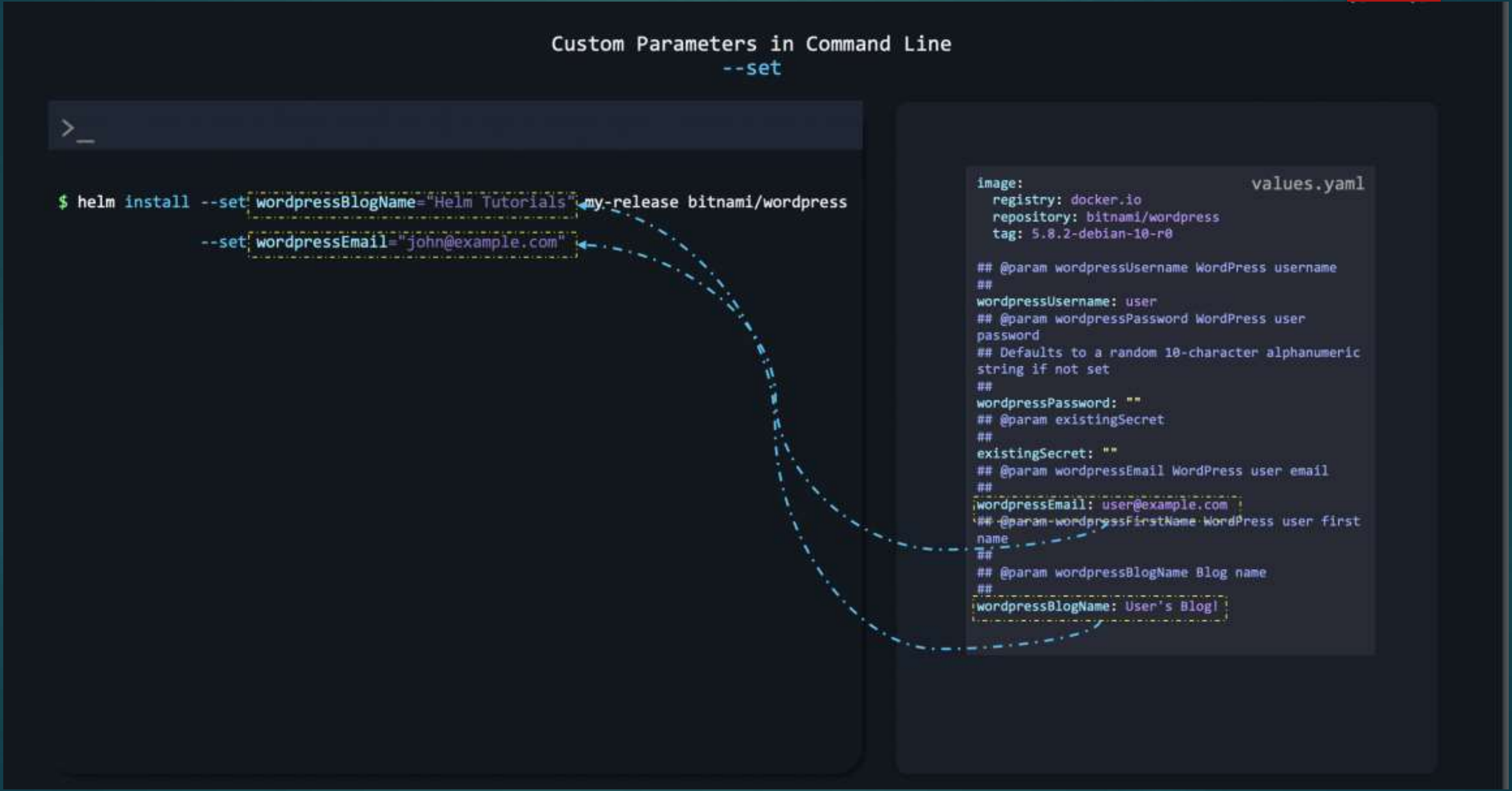


```
>_
```

```
$ helm install my-release bitnami/wordpress
```

values.yaml

```yaml
image:
  registry: docker.io
  repository: bitnami/wordpress
  tag: 5.8.2-debian-10-r0

## @param wordpressUsername WordPress username
##
wordpressUsername: user
## @param wordpressPassword WordPress user
password
## Defaults to a random 10-character alphanumeric
string if not set
##
wordpressPassword: ""
## @param existingSecret
##
existingSecret: ""
## @param wordpressEmail WordPress user email
##
wordpressEmail: user@example.com
## @param wordpressFirstName WordPress user first
name
##
## @param wordpressBlogName Blog name
##
wordpressBlogName: User's Blog!
```

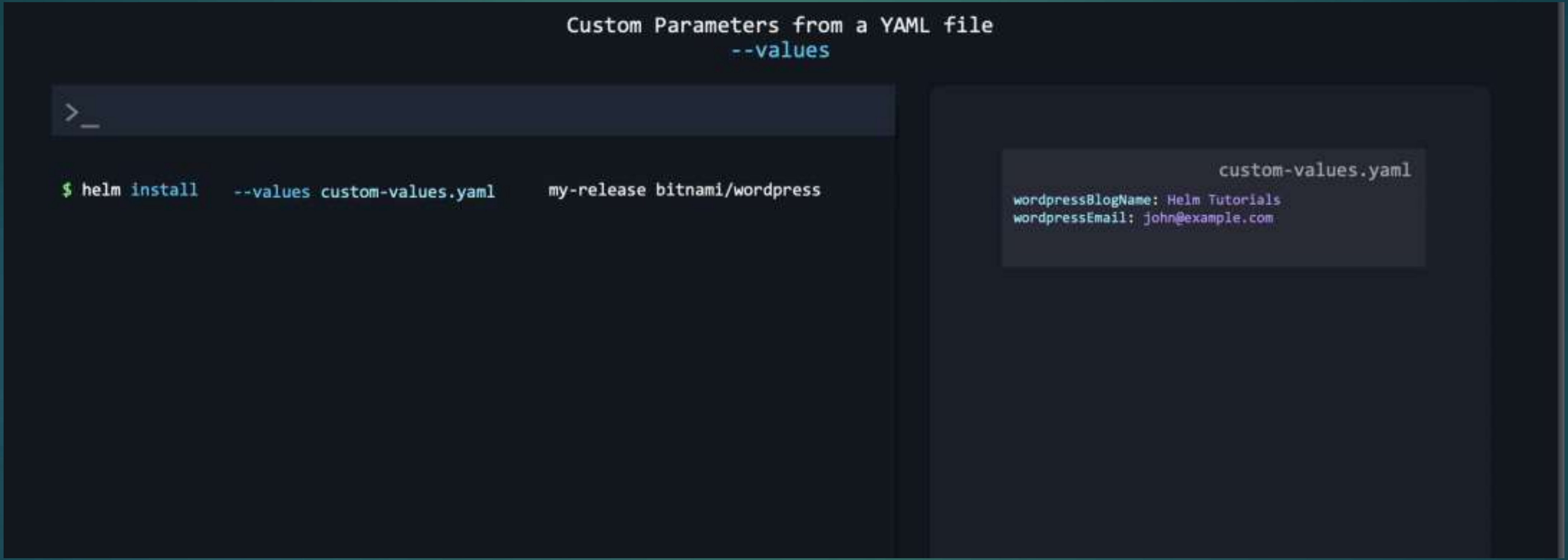


Custom Parameters in Command Line
--set

```
$ helm install --set wordpressBlogName="Helm Tutorials" my-release bitnami/wordpress
               --set wordpressEmail="john@example.com"
```

values.yaml

```yaml
image:
  registry: docker.io
  repository: bitnami/wordpress
  tag: 5.8.2-debian-10-r0

## @param wordpressUsername WordPress username
##
wordpressUsername: user
## @param wordpressPassword WordPress user
password
## Defaults to a random 10-character alphanumeric
string if not set
##
wordpressPassword: ""
## @param existingSecret
##
existingSecret: ""
## @param wordpressEmail WordPress user email
##
wordpressEmail: user@example.com
## @param wordpressFirstName WordPress user first
name
##
## @param wordpressBlogName Blog name
##
wordpressBlogName: User's Blog!
```

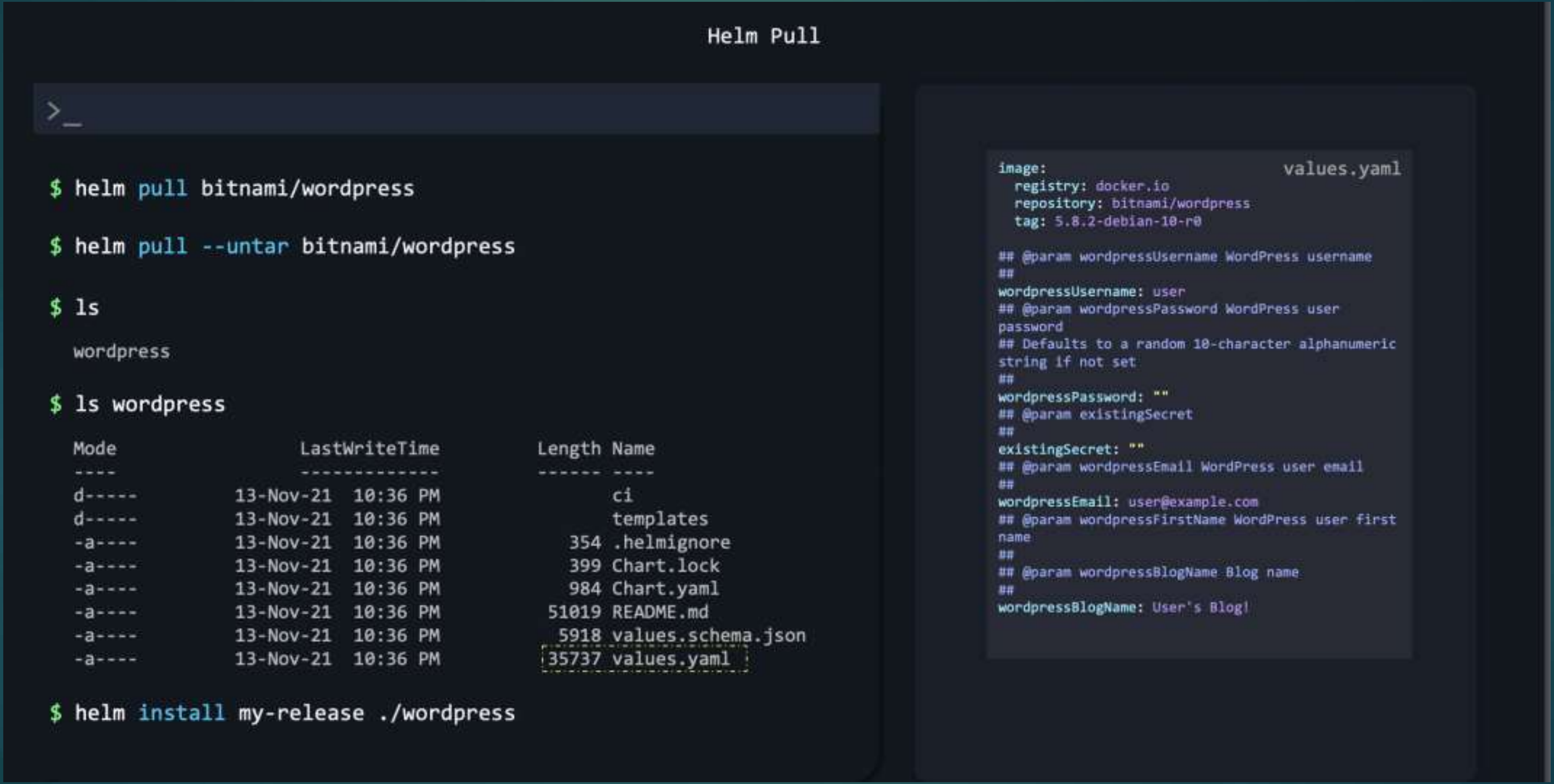


Custom Parameters from a YAML file
--values

```
$ helm install   --values custom-values.yaml    my-release bitnami/wordpress
```

custom-values.yaml

```yaml
wordpressBlogName: Helm Tutorials
wordpressEmail: john@example.com
```

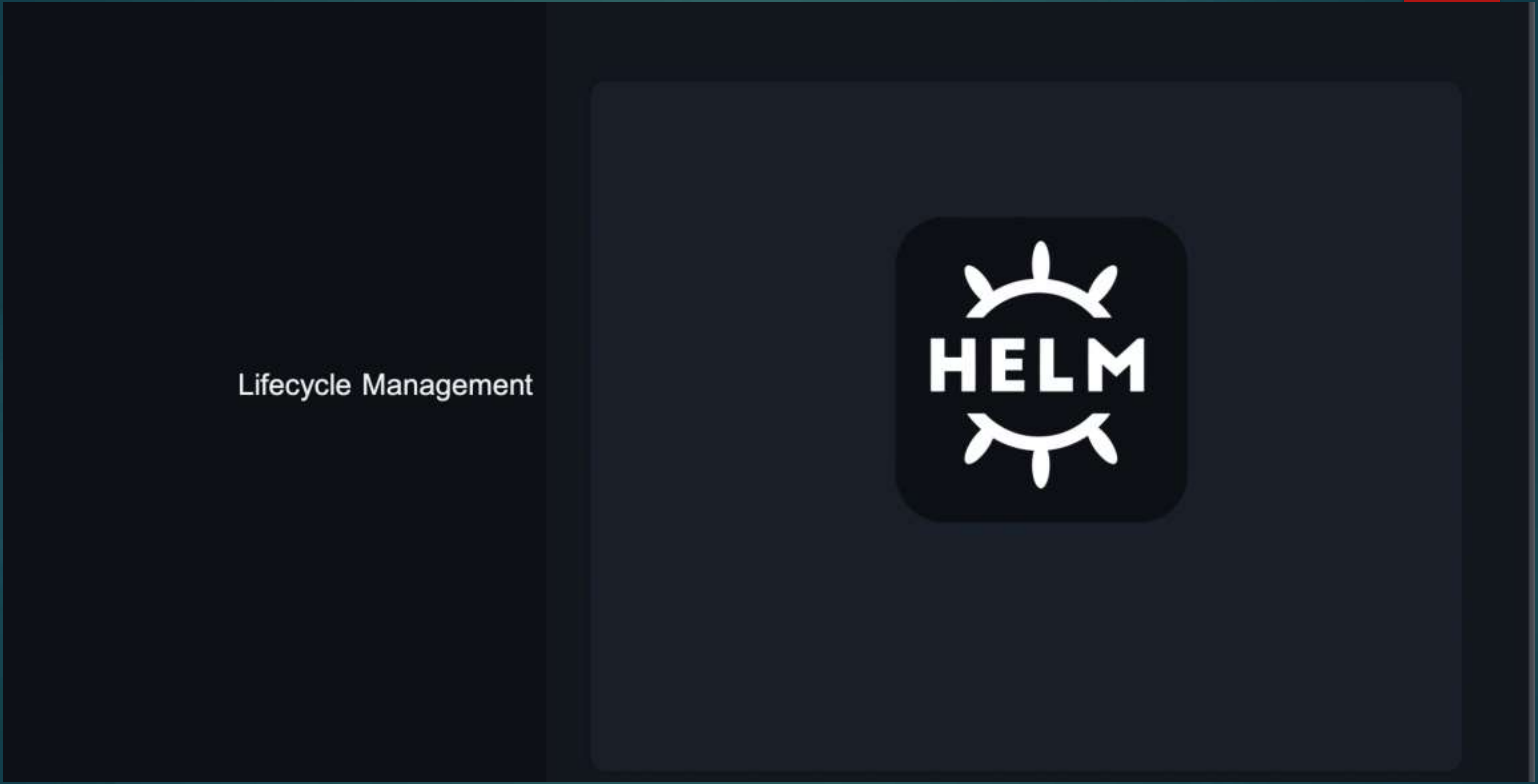


Helm Pull

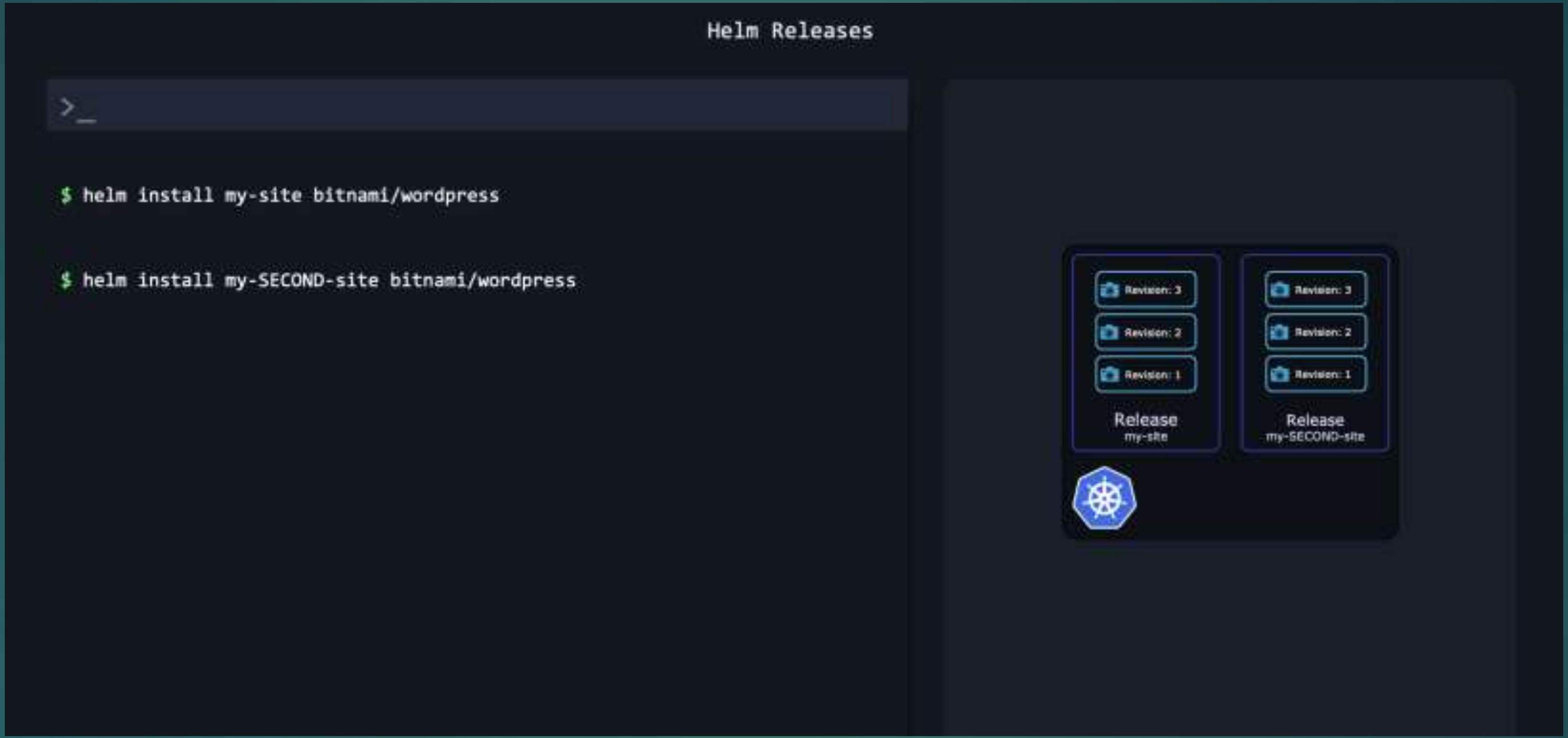
```
>_

$ helm pull bitnami/wordpress

$ helm pull --untar bitnami/wordpress

$ ls
  wordpress

$ ls wordpress

  Mode                LastWriteTime         Length Name
  ----                -------------         ------ ----
  d-----        13-Nov-21  10:36 PM                ci
  d-----        13-Nov-21  10:36 PM                templates
  -a----        13-Nov-21  10:36 PM            354 .helmignore
  -a----        13-Nov-21  10:36 PM            399 Chart.lock
  -a----        13-Nov-21  10:36 PM            984 Chart.yaml
  -a----        13-Nov-21  10:36 PM          51019 README.md
  -a----        13-Nov-21  10:36 PM           5918 values.schema.json
  -a----        13-Nov-21  10:36 PM          35737 values.yaml

$ helm install my-release ./wordpress
```

values.yaml

```
image:
  registry: docker.io
  repository: bitnami/wordpress
  tag: 5.8.2-debian-10-r0

## @param wordpressUsername WordPress username
##
wordpressUsername: user
## @param wordpressPassword WordPress user
password
## Defaults to a random 10-character alphanumeric
string if not set
##
wordpressPassword: ""
## @param existingSecret
##
existingSecret: ""
## @param wordpressEmail WordPress user email
##
wordpressEmail: user@example.com
## @param wordpressFirstName WordPress user first
name
##
## @param wordpressBlogName Blog name
##
wordpressBlogName: User's Blog!
```

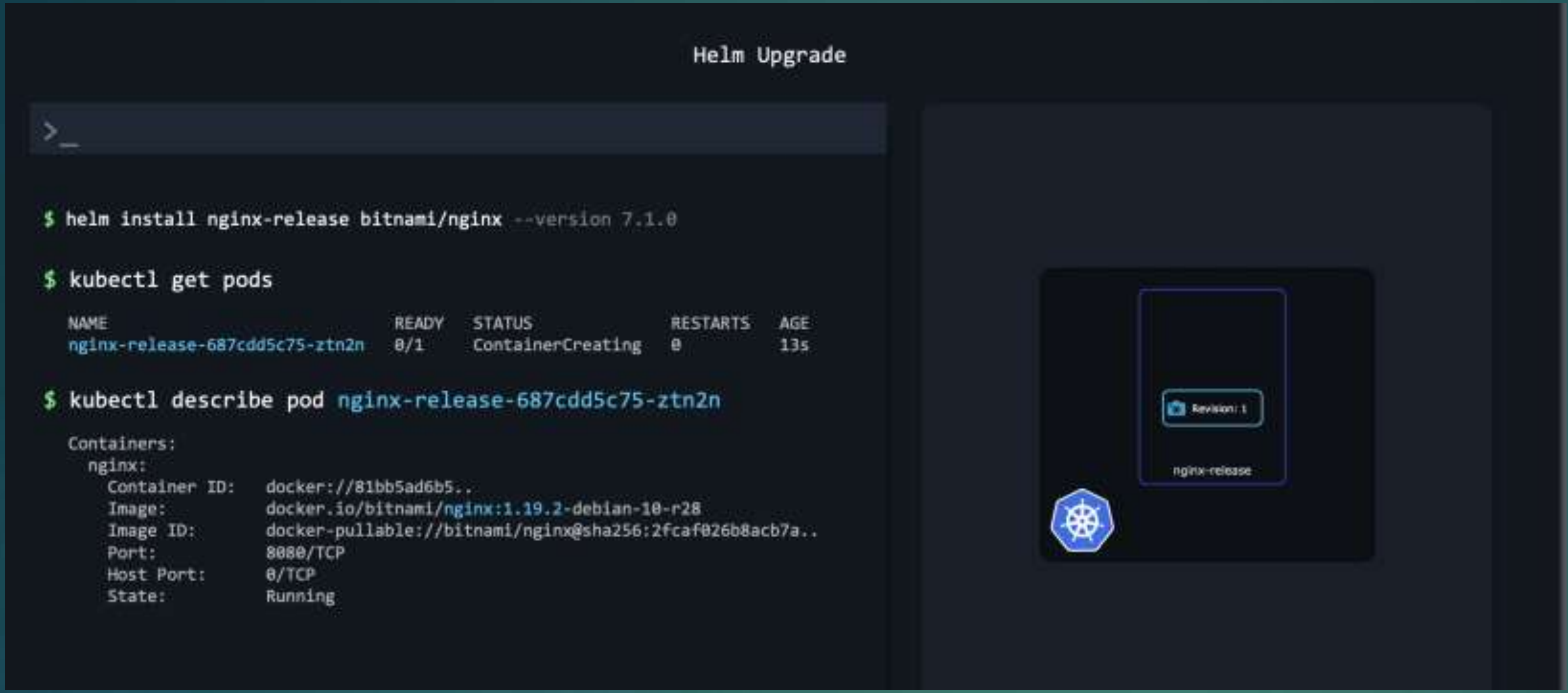


Lifecycle Management



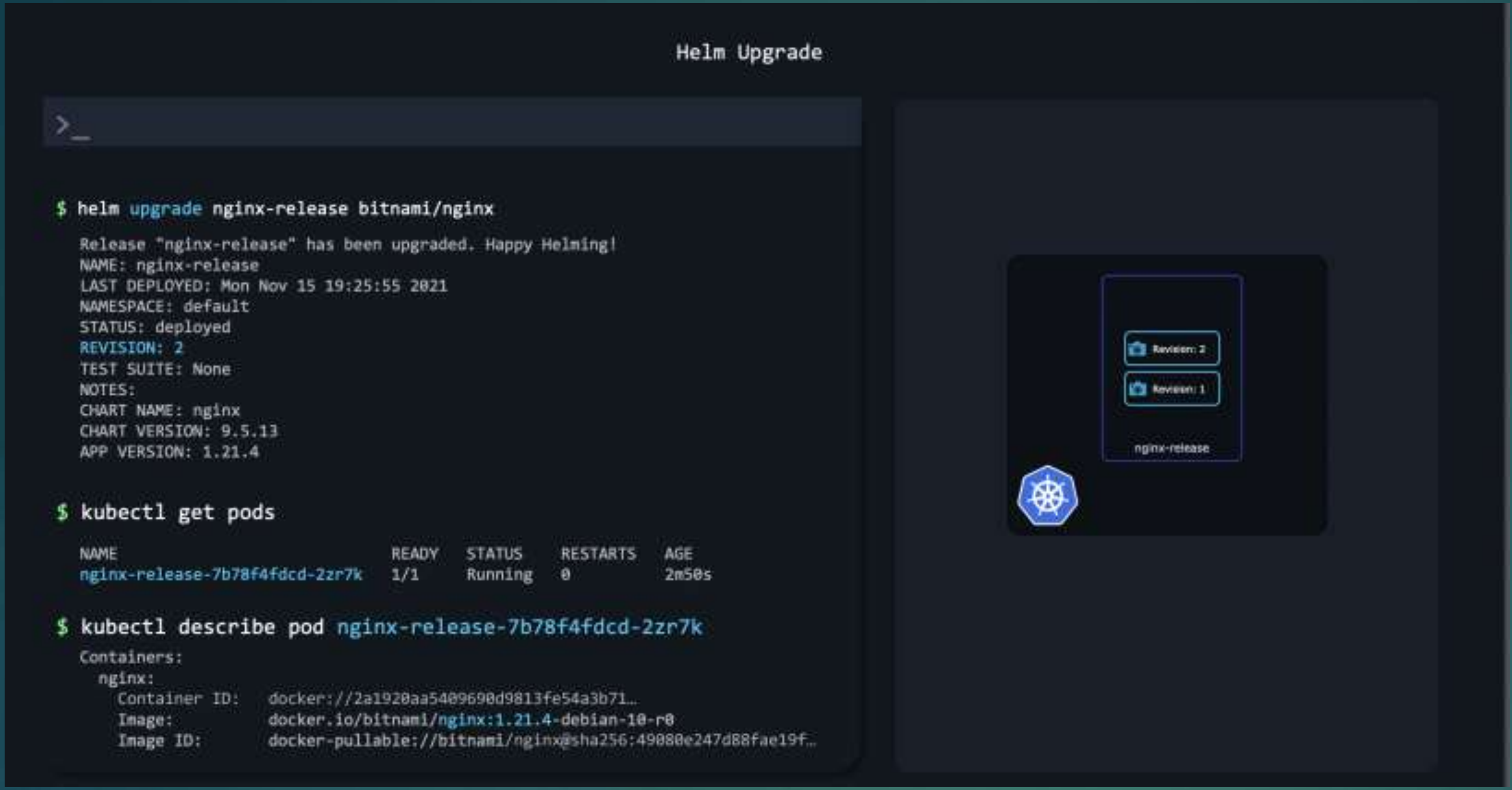
Helm Releases

```
>_

$ helm install my-site bitnami/wordpress

$ helm install my-SECOND-site bitnami/wordpress
```

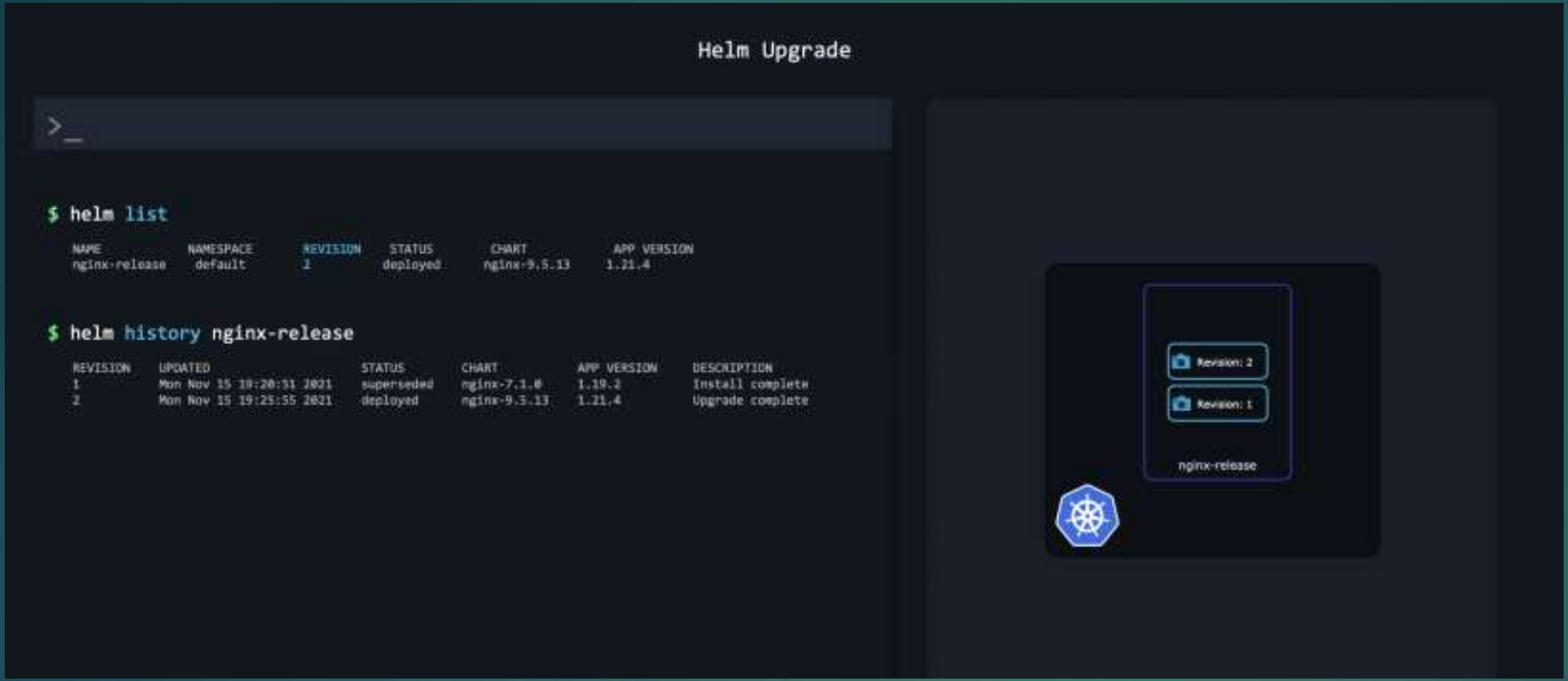


Helm Upgrade

```
>_

$ helm install nginx-release bitnami/nginx --version 7.1.0

$ kubectl get pods

  NAME                             READY   STATUS              RESTARTS   AGE
  nginx-release-687cdd5c75-ztn2n   0/1     ContainerCreating   0          13s

$ kubectl describe pod nginx-release-687cdd5c75-ztn2n

  Containers:
    nginx:
      Container ID:   docker://81bb5ad6b5..
      Image:          docker.io/bitnami/nginx:1.19.2-debian-10-r28
      Image ID:       docker-pullable://bitnami/nginx@sha256:2fcaf026b8acb7a..
      Port:           8080/TCP
      Host Port:      0/TCP
      State:          Running
```

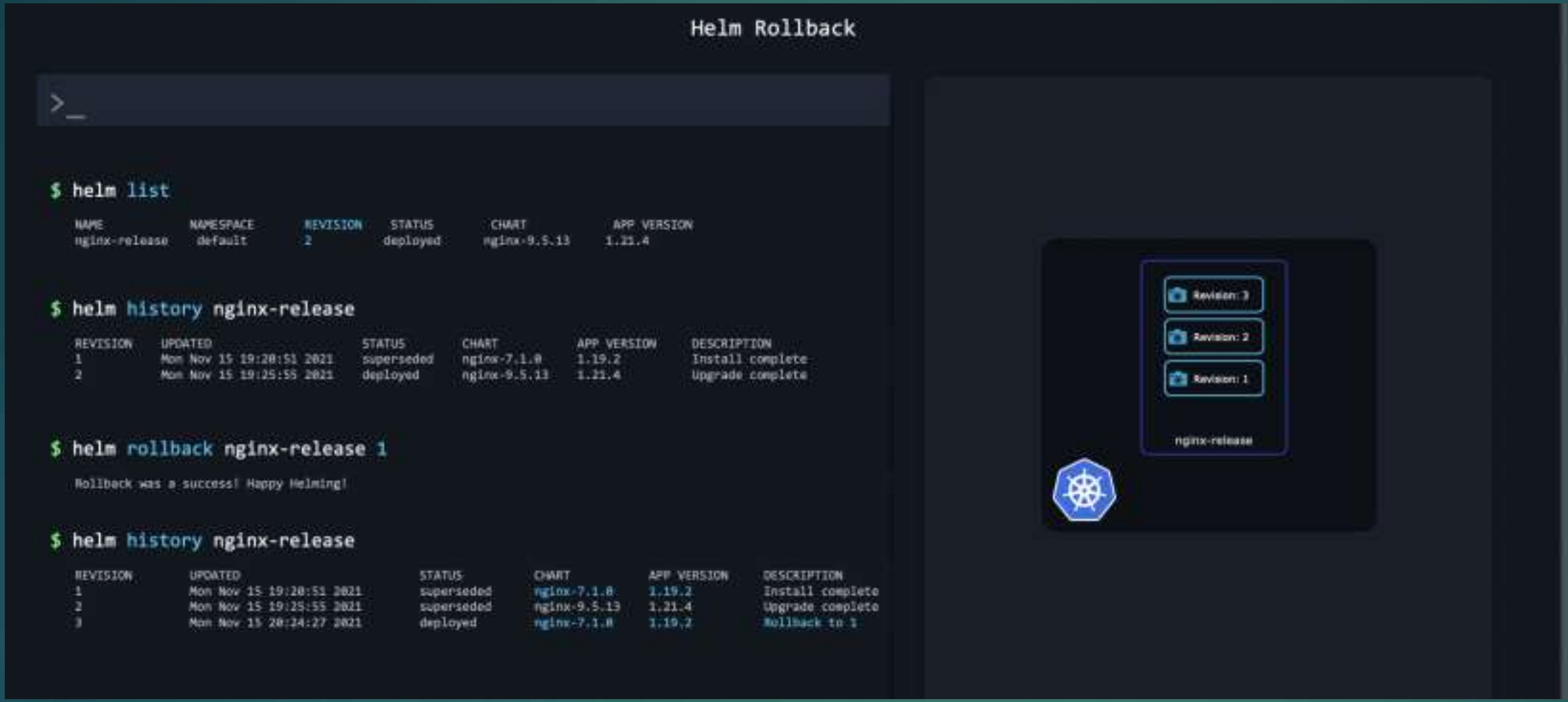


```
$ helm upgrade nginx-release bitnami/nginx
  Release "nginx-release" has been upgraded. Happy Helming!
  NAME: nginx-release
  LAST DEPLOYED: Mon Nov 15 19:25:55 2021
  NAMESPACE: default
  STATUS: deployed
  REVISION: 2
  TEST SUITE: None
  NOTES:
  CHART NAME: nginx
  CHART VERSION: 9.5.13
  APP VERSION: 1.21.4

$ kubectl get pods
  NAME                             READY   STATUS    RESTARTS   AGE
  nginx-release-7b78f4fdcd-2zr7k   1/1     Running   0          2m50s

$ kubectl describe pod nginx-release-7b78f4fdcd-2zr7k
  Containers:
    nginx:
      Container ID:   docker://2a1920aa5409690d9813fe54a3b71…
      Image:          docker.io/bitnami/nginx:1.21.4-debian-10-r0
      Image ID:       docker-pullable://bitnami/nginx@sha256:49080e247d88fae19f…
```



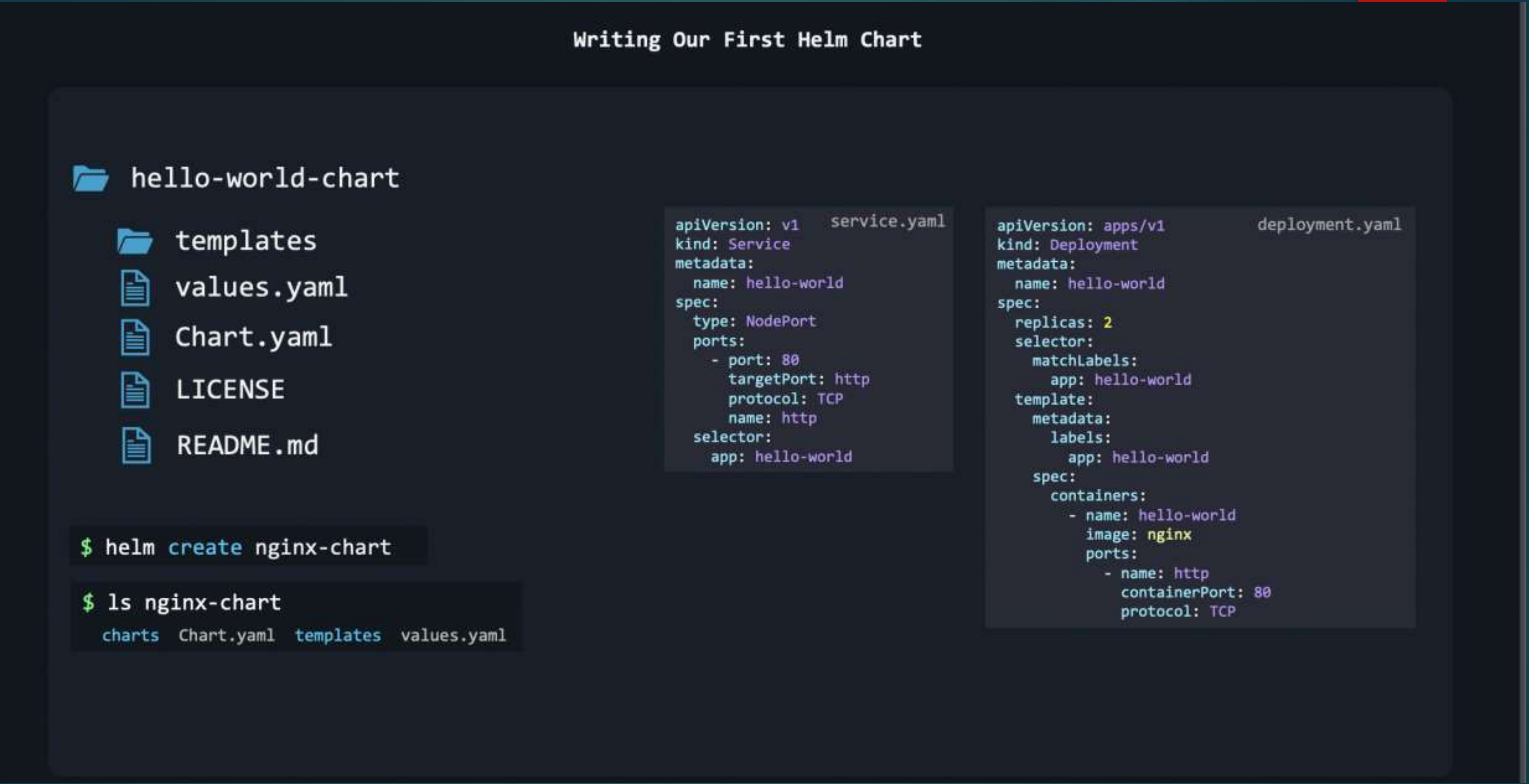
Helm Upgrade

```
$ helm list

   NAME            NAMESPACE      REVISION   STATUS      CHART          APP VERSION
   nginx-release    default       2          deployed    nginx-9.5.13   1.21.4

$ helm history nginx-release

   REVISION   UPDATED                    STATUS       CHART          APP VERSION   DESCRIPTION
   1          Mon Nov 15 19:20:51 2021   superseded   nginx-7.1.0    1.19.2        Install complete
   2          Mon Nov 15 19:25:55 2021   deployed     nginx-9.5.13   1.21.4        Upgrade complete
```



Helm Rollback

```
>_

$ helm list

  NAME            NAMESPACE     REVISION  STATUS      CHART          APP VERSION
  nginx-release    default      2         deployed    nginx-9.5.13   1.21.4

$ helm history nginx-release

  REVISION   UPDATED                    STATUS       CHART          APP VERSION    DESCRIPTION
  1          Mon Nov 15 19:20:51 2021   superseded   nginx-7.1.0    1.19.2         Install complete
  2          Mon Nov 15 19:25:55 2021   deployed     nginx-9.5.13   1.21.4         Upgrade complete

$ helm rollback nginx-release 1

  Rollback was a success! Happy Helming!

$ helm history nginx-release

  REVISION      UPDATED                        STATUS        CHART          APP VERSION    DESCRIPTION
  1             Mon Nov 15 19:20:51 2021       superseded    nginx-7.1.0    1.19.2         Install complete
  2             Mon Nov 15 19:25:55 2021       superseded    nginx-9.5.13   1.21.4         Upgrade complete
  3             Mon Nov 15 20:24:27 2021       deployed      nginx-7.1.0    1.19.2         Rollback to 1
```



Writing Our First Helm Chart

```
hello-world-chart
    templates
    values.yaml
    Chart.yaml
    LICENSE
    README.md
```

```
$ helm create nginx-chart
```

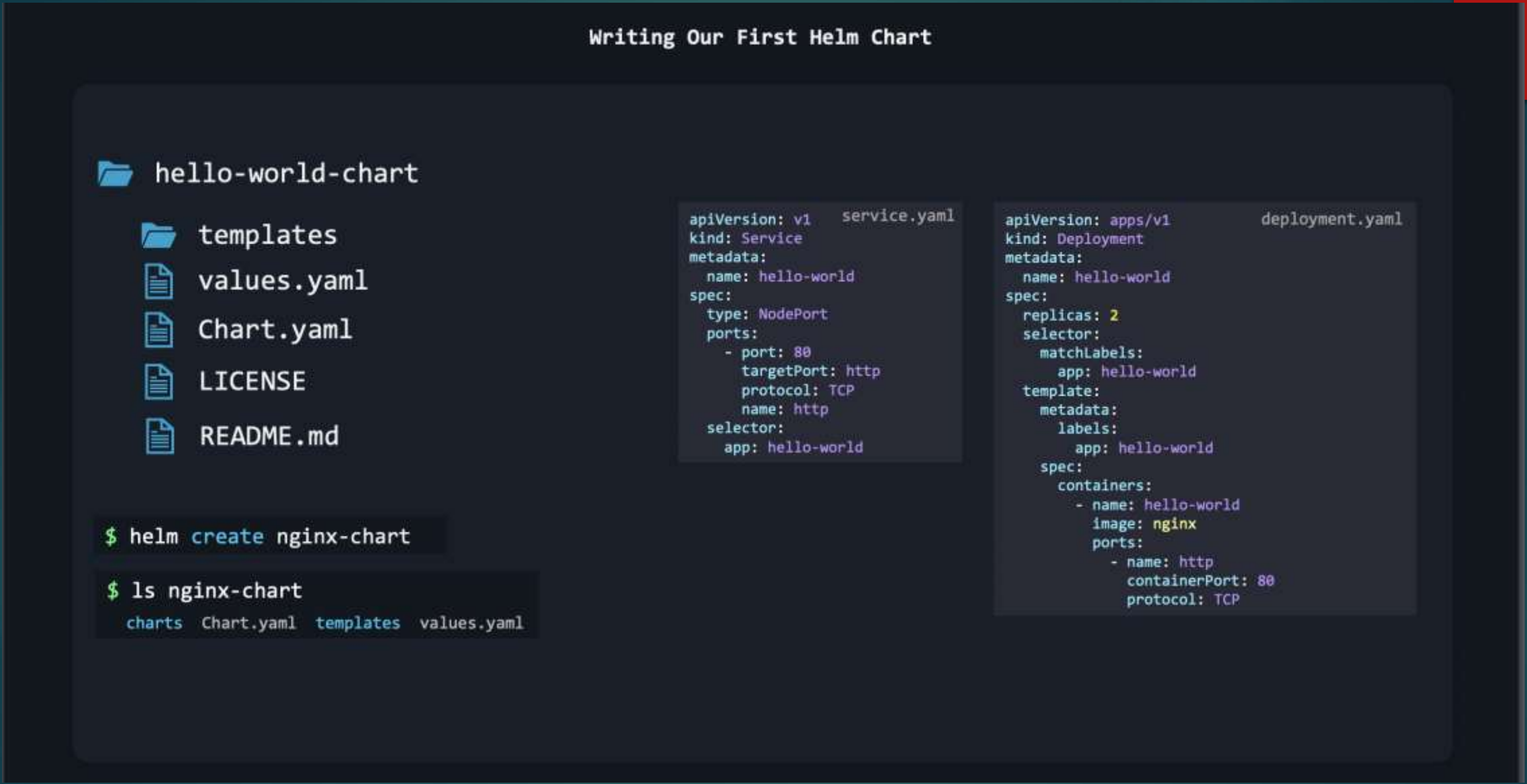
```
$ ls nginx-chart
  charts  Chart.yaml  templates  values.yaml
```

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-world
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: nginx
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```



Writing Our First Helm Chart

- hello-world-chart
  - templates
  - values.yaml
  - Chart.yaml
  - LICENSE
  - README.md

```
$ helm create nginx-chart
```

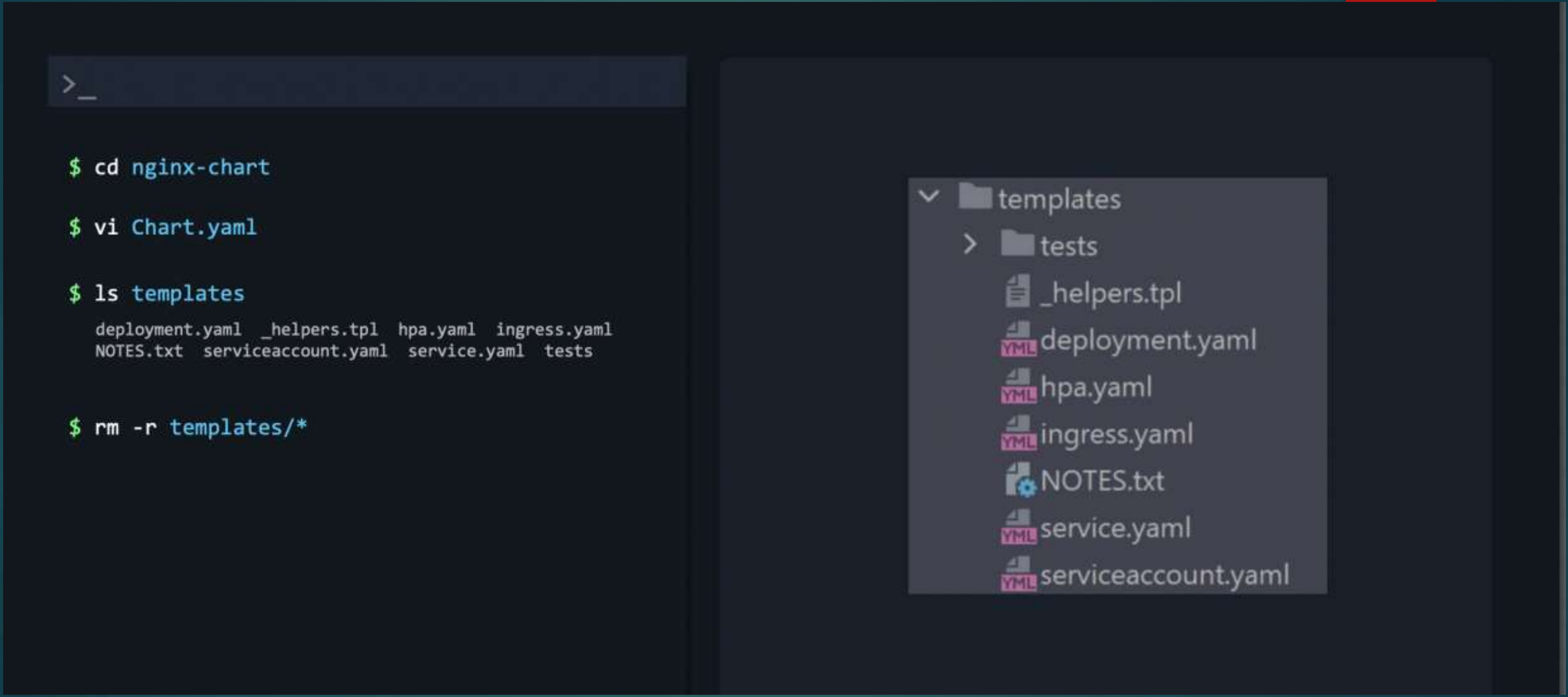
```
$ ls nginx-chart
  charts  Chart.yaml  templates  values.yaml
```

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-world
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

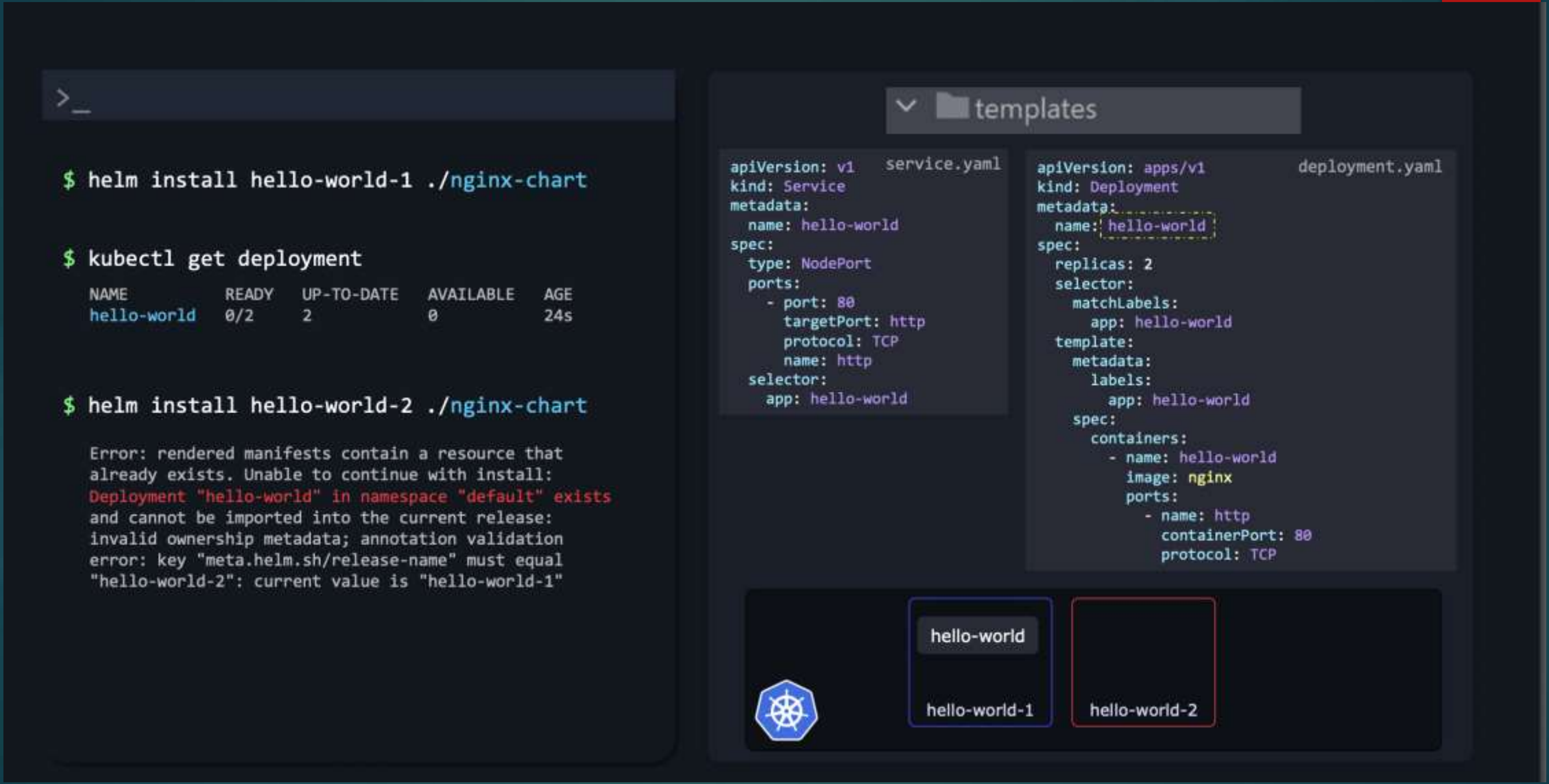
deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: nginx
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```



```
>_

$ cd nginx-chart

$ vi Chart.yaml

$ ls templates
  deployment.yaml  _helpers.tpl  hpa.yaml  ingress.yaml
  NOTES.txt  serviceaccount.yaml  service.yaml  tests


$ rm -r templates/*
```

```
templates
  tests
  _helpers.tpl
  deployment.yaml
  hpa.yaml
  ingress.yaml
  NOTES.txt
  service.yaml
  serviceaccount.yaml
```



```
>_

$ helm install hello-world-1 ./nginx-chart

$ kubectl get deployment
  NAME          READY   UP-TO-DATE   AVAILABLE   AGE
  hello-world   0/2     2            0           24s


$ helm install hello-world-2 ./nginx-chart

  Error: rendered manifests contain a resource that
  already exists. Unable to continue with install:
  Deployment "hello-world" in namespace "default" exists
  and cannot be imported into the current release:
  invalid ownership metadata; annotation validation
  error: key "meta.helm.sh/release-name" must equal
  "hello-world-2": current value is "hello-world-1"
```

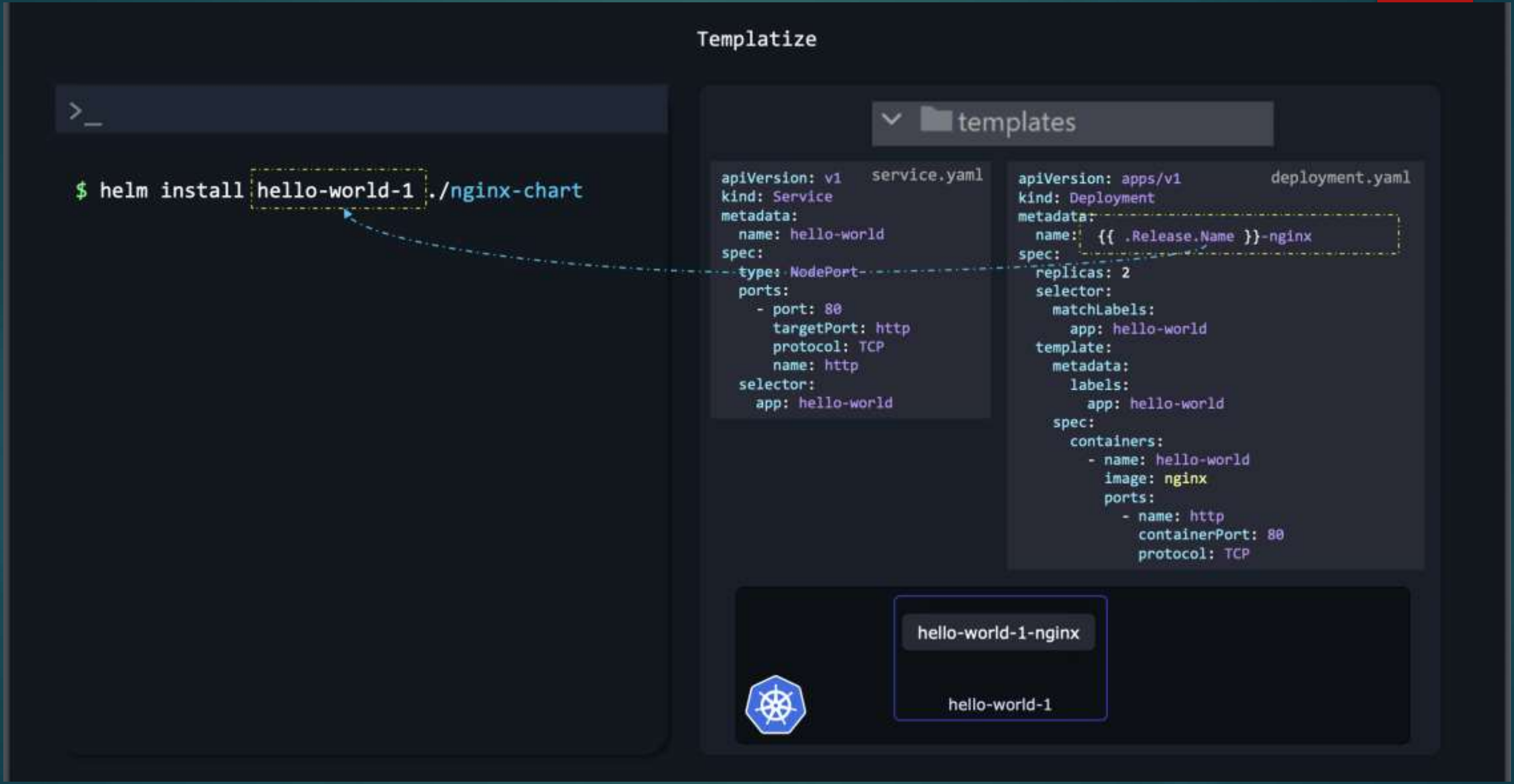
templates

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-world
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: nginx
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```



Templatize

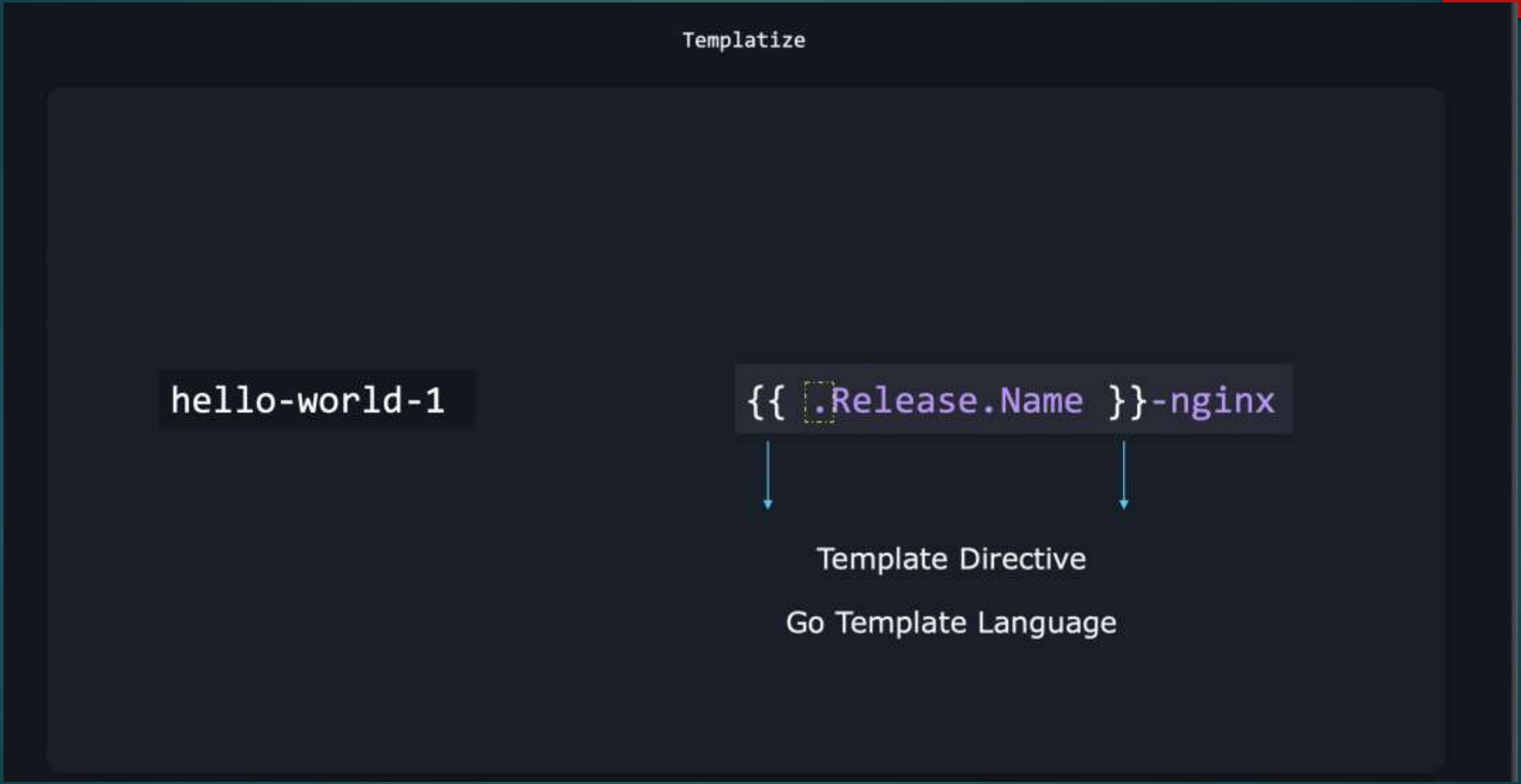
```
$ helm install hello-world-1 ./nginx-chart
```

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-world
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: nginx
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```



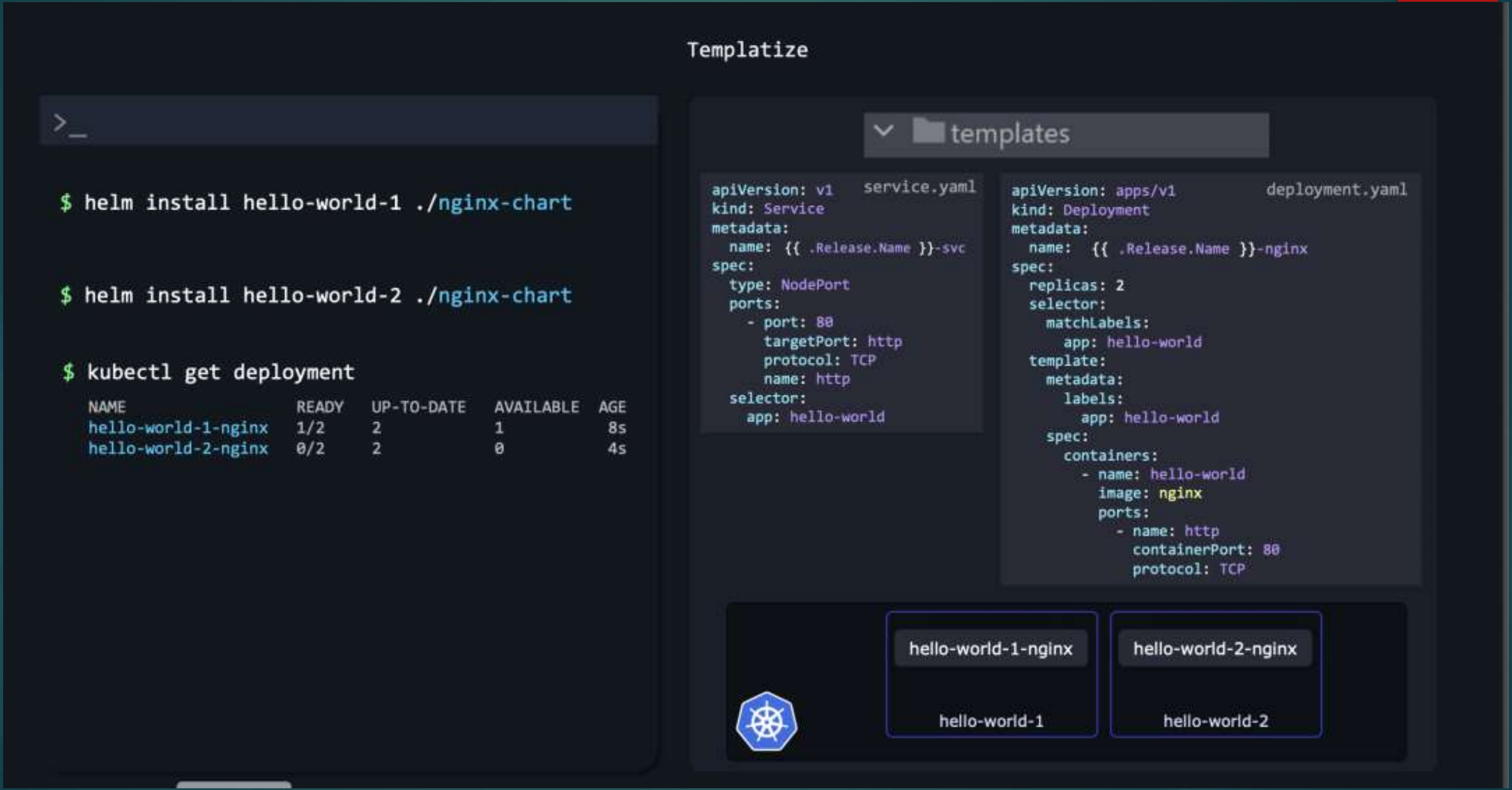
Templatize

```
hello-world-1
```

```
{{ .Release.Name }}-nginx
```

Template Directive

Go Template Language



Templatize

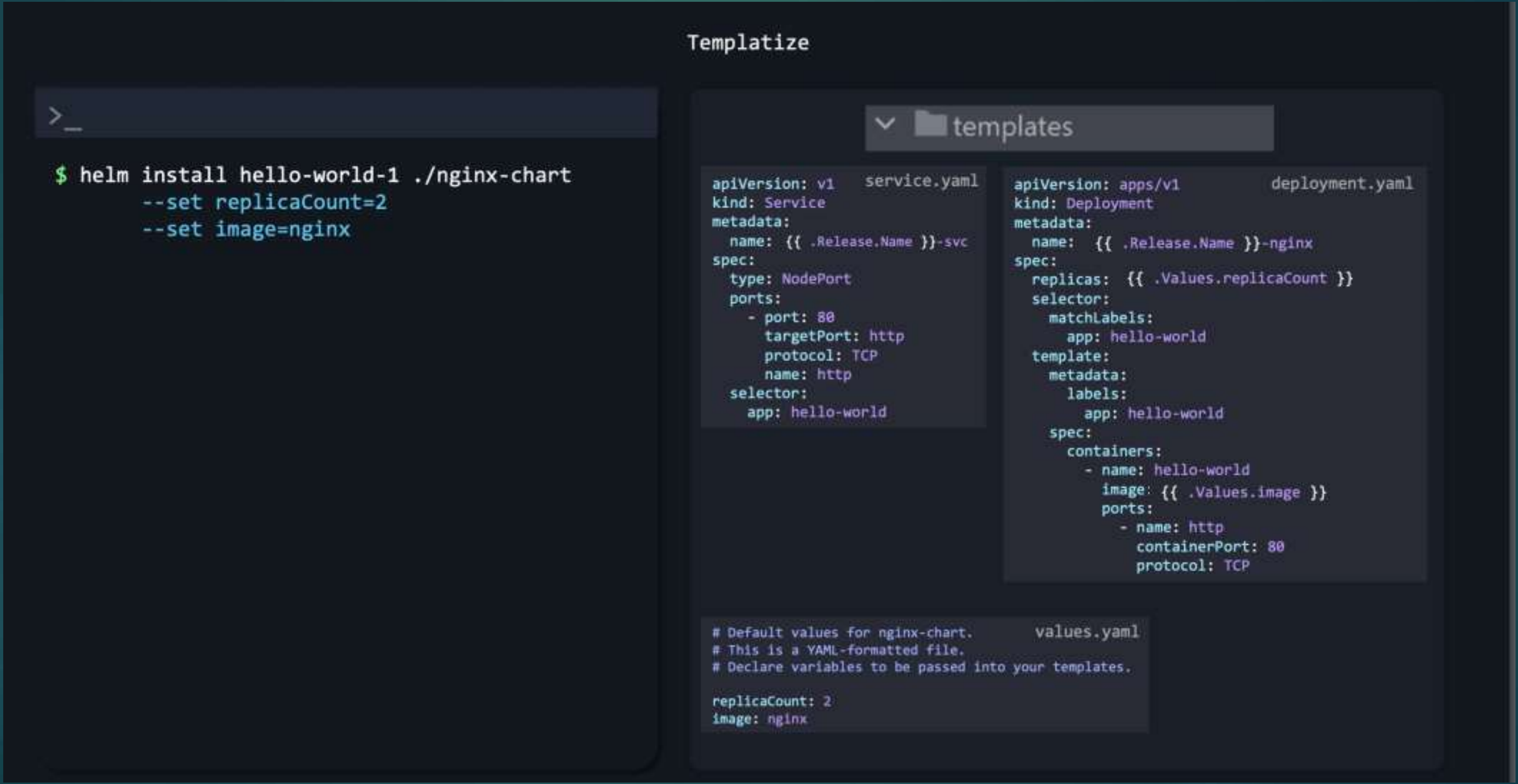
```
$ helm install hello-world-1 ./nginx-chart

$ helm install hello-world-2 ./nginx-chart

$ kubectl get deployment
  NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
  hello-world-1-nginx  1/2     2            1            8s
  hello-world-2-nginx  0/2     2            0            4s
```

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-svc
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name:  {{ .Release.Name }}-nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: nginx
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```



Templatize

```
$ helm install hello-world-1 ./nginx-chart
      --set replicaCount=2
      --set image=nginx
```

service.yaml

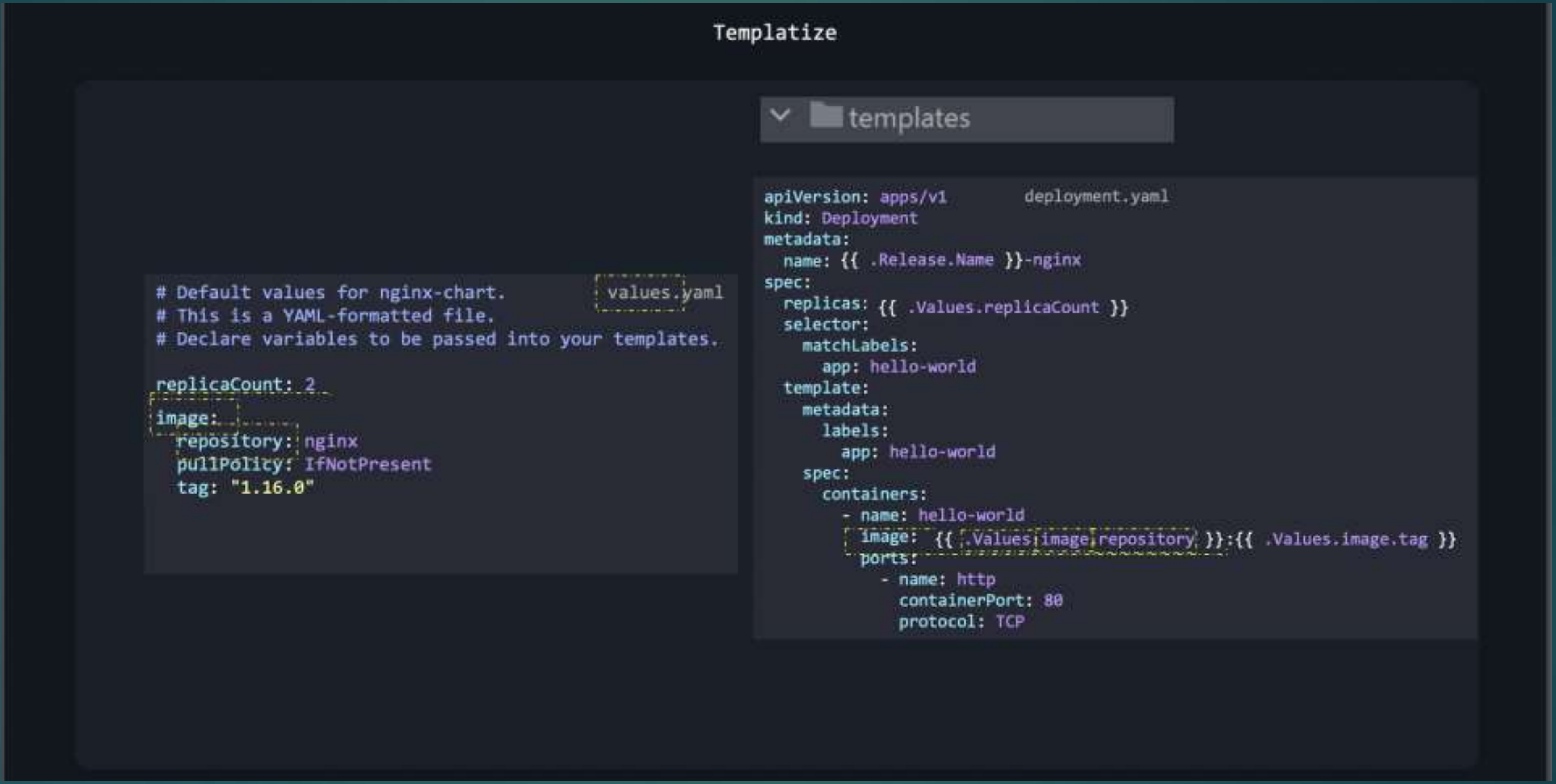
```yaml
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-svc
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name:  {{ .Release.Name }}-nginx
spec:
  replicas:  {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ .Values.image }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

values.yaml

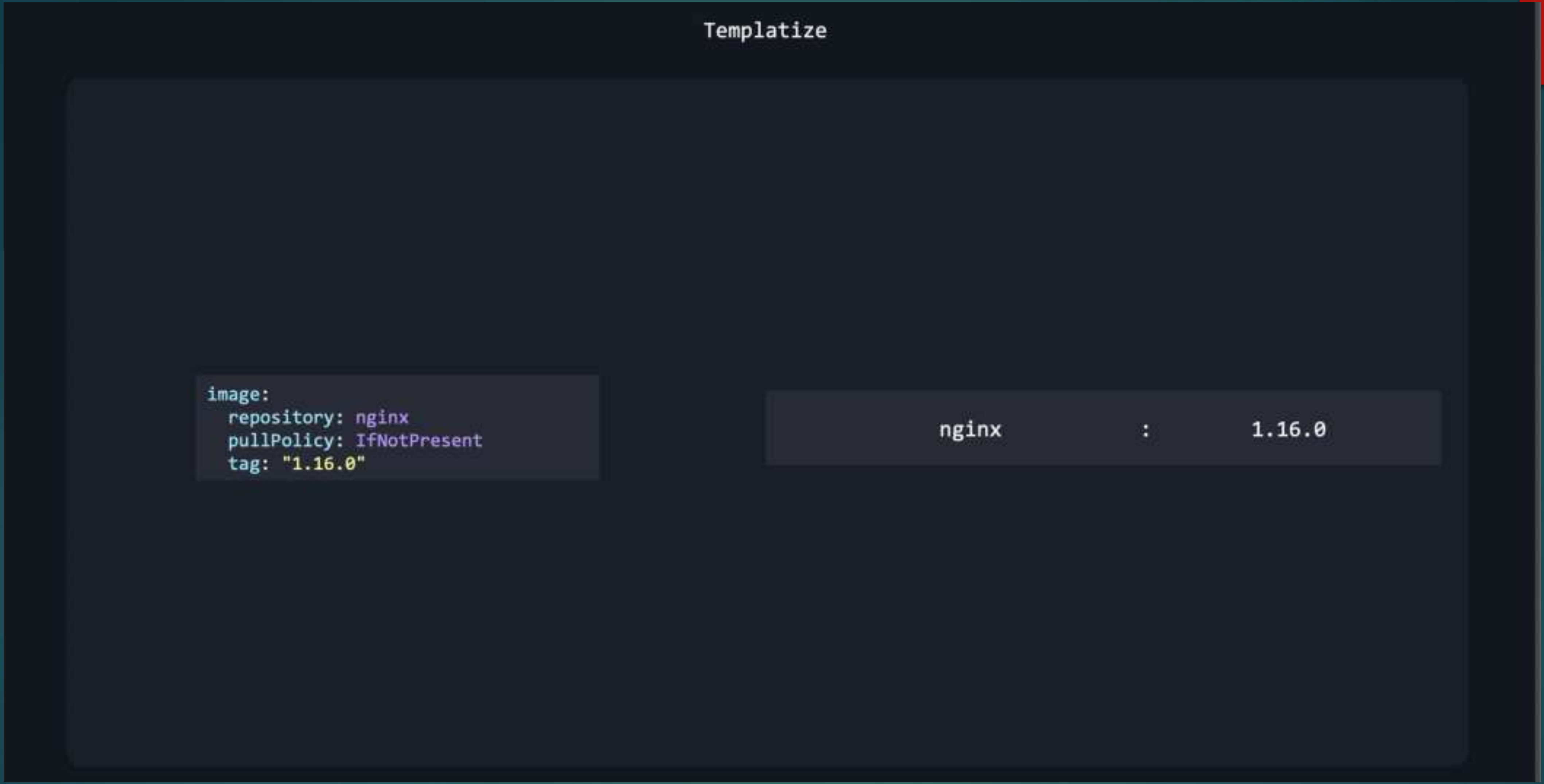
```yaml
# Default values for nginx-chart.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicaCount: 2
image: nginx
```



Templatize

```yaml
# Default values for nginx-chart.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicaCount: 2
image:
  repository: nginx
  pullPolicy: IfNotPresent
  tag: "1.16.0"
```

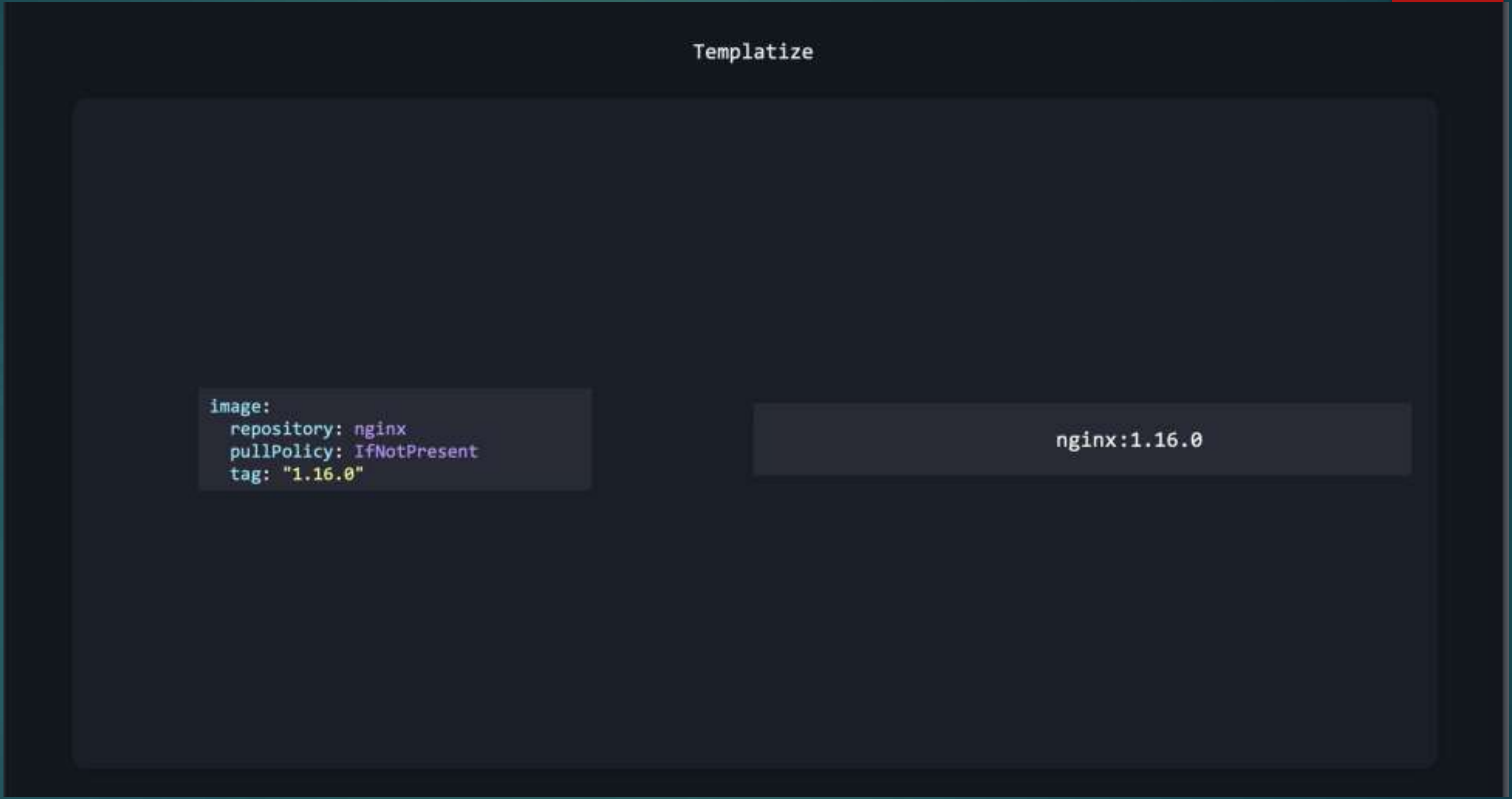
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ .Values.image.repository }}:{{ .Values.image.tag }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```



Templatize

```yaml
image:
  repository: nginx
  pullPolicy: IfNotPresent
  tag: "1.16.0"
```

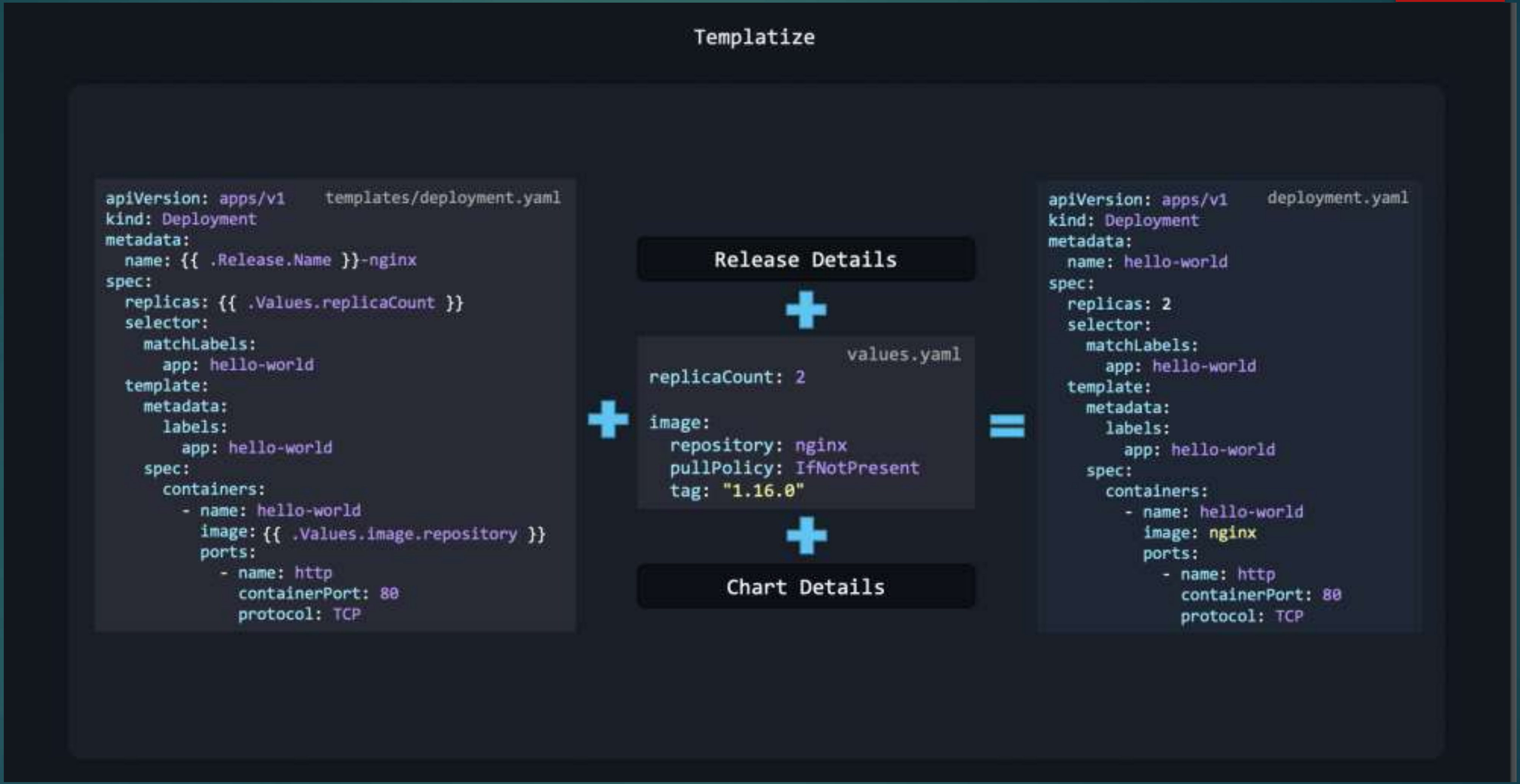
```
nginx        :        1.16.0
```



Templatize

```yaml
image:
  repository: nginx
  pullPolicy: IfNotPresent
  tag: "1.16.0"
```

```
nginx:1.16.0
```



Templatize

templates/deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ .Values.image.repository }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

+

Release Details

+

values.yaml

```
replicaCount: 2

image:
  repository: nginx
  pullPolicy: IfNotPresent
  tag: "1.16.0"
```

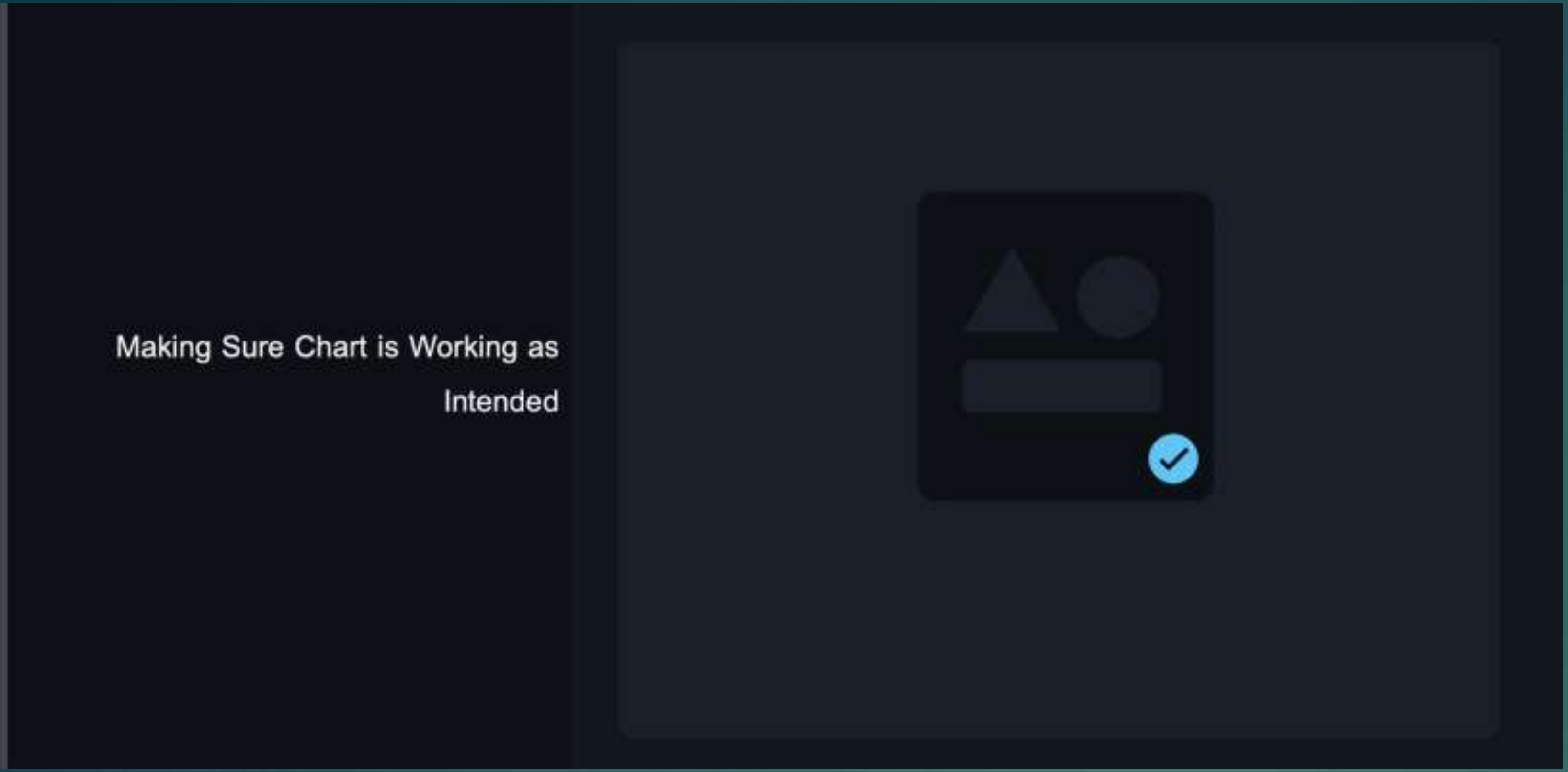
+

Chart Details

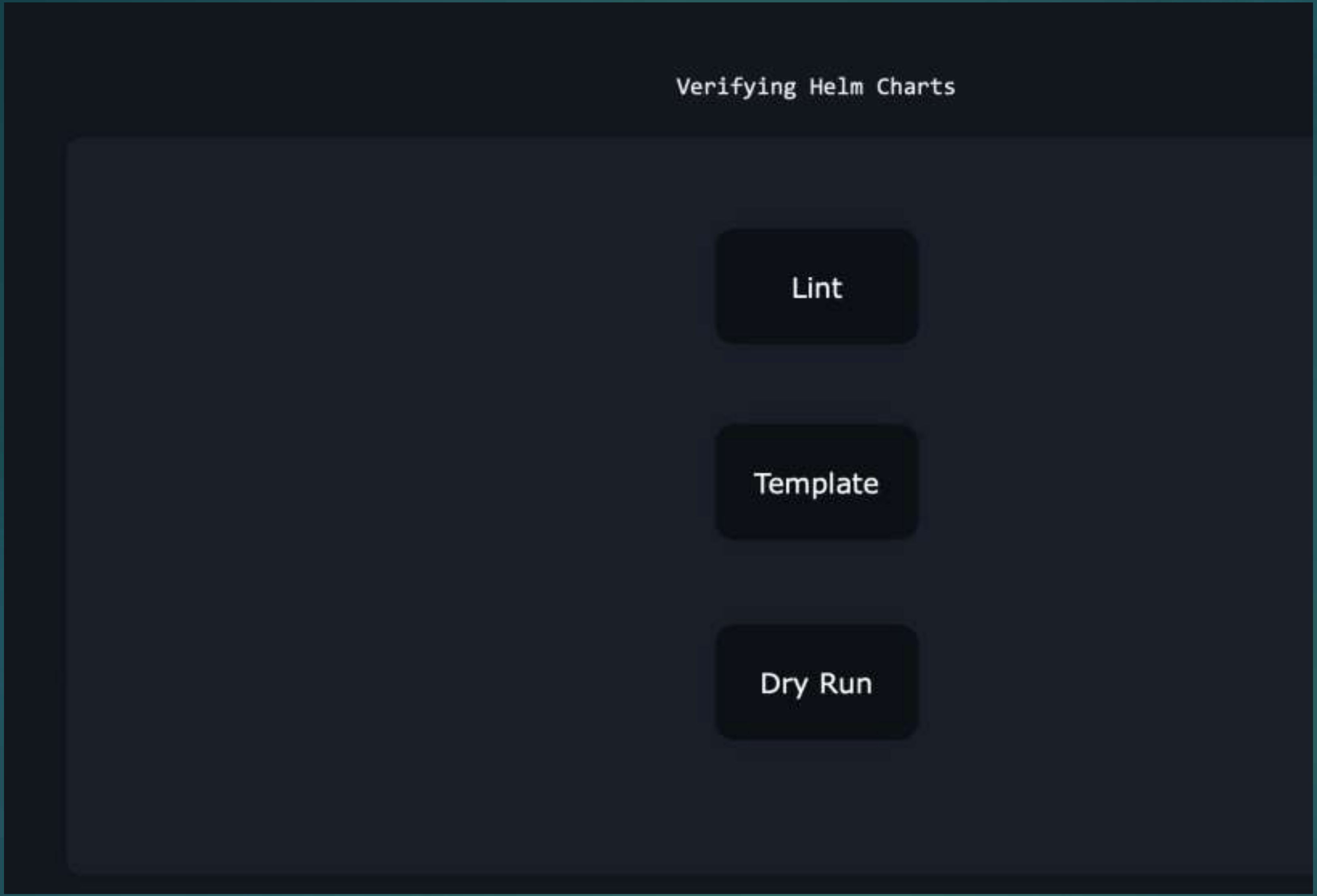
=

deployment.yaml

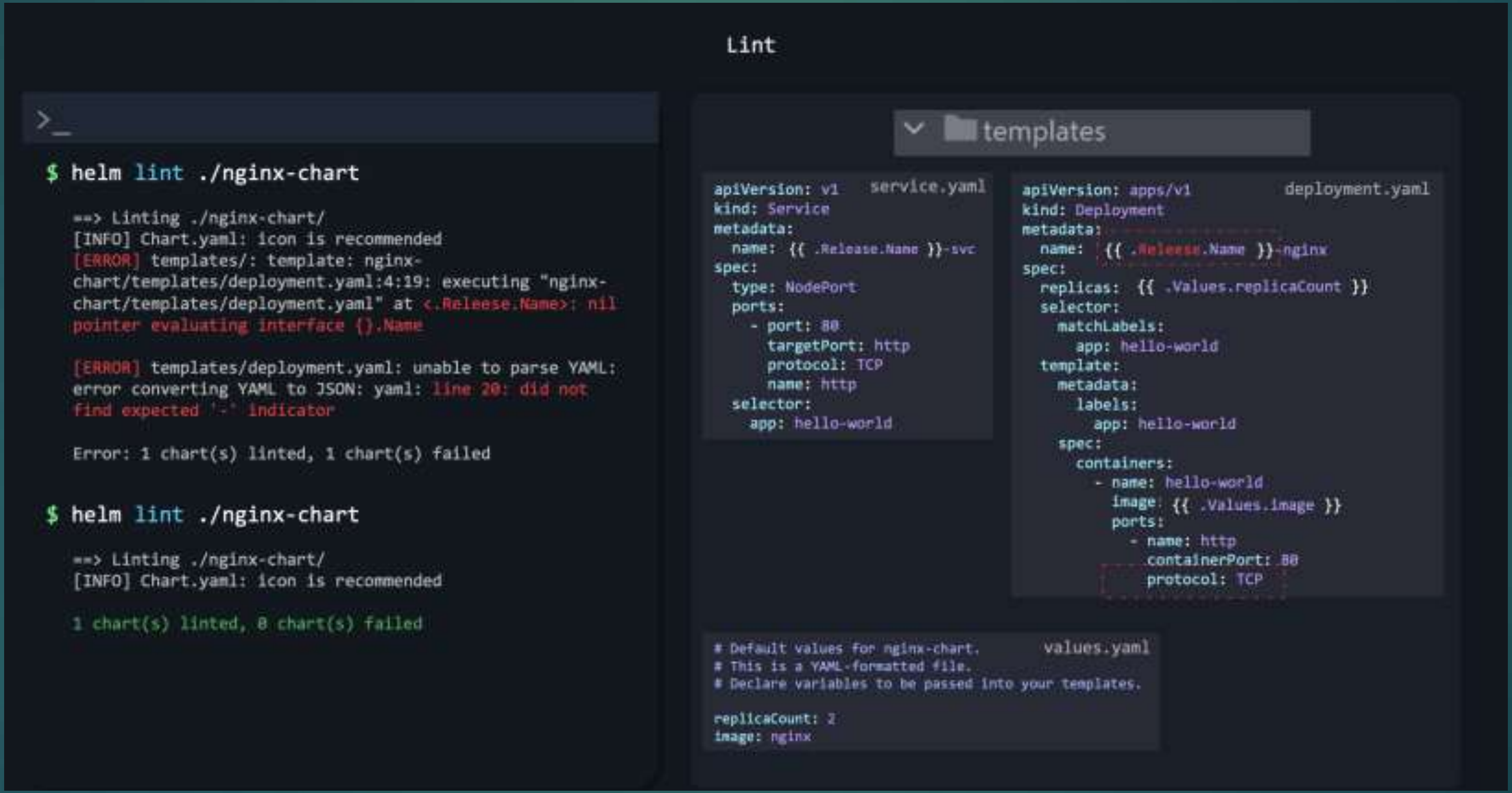
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: nginx
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```



Making Sure Chart is Working as Intended


Verifying Helm Charts
Lint
Template
Dry Run

1-lint



Lint

```
>_

$ helm lint ./nginx-chart

  ==> Linting ./nginx-chart/
  [INFO] Chart.yaml: icon is recommended
  [ERROR] templates/: template: nginx-
  chart/templates/deployment.yaml:4:19: executing "nginx-
  chart/templates/deployment.yaml" at <.Releese.Name>: nil
  pointer evaluating interface {}.Name

  [ERROR] templates/deployment.yaml: unable to parse YAML:
  error converting YAML to JSON: yaml: line 20: did not
  find expected '-' indicator

  Error: 1 chart(s) linted, 1 chart(s) failed

$ helm lint ./nginx-chart

  ==> Linting ./nginx-chart/
  [INFO] Chart.yaml: icon is recommended

  1 chart(s) linted, 0 chart(s) failed
```

service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-svc
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

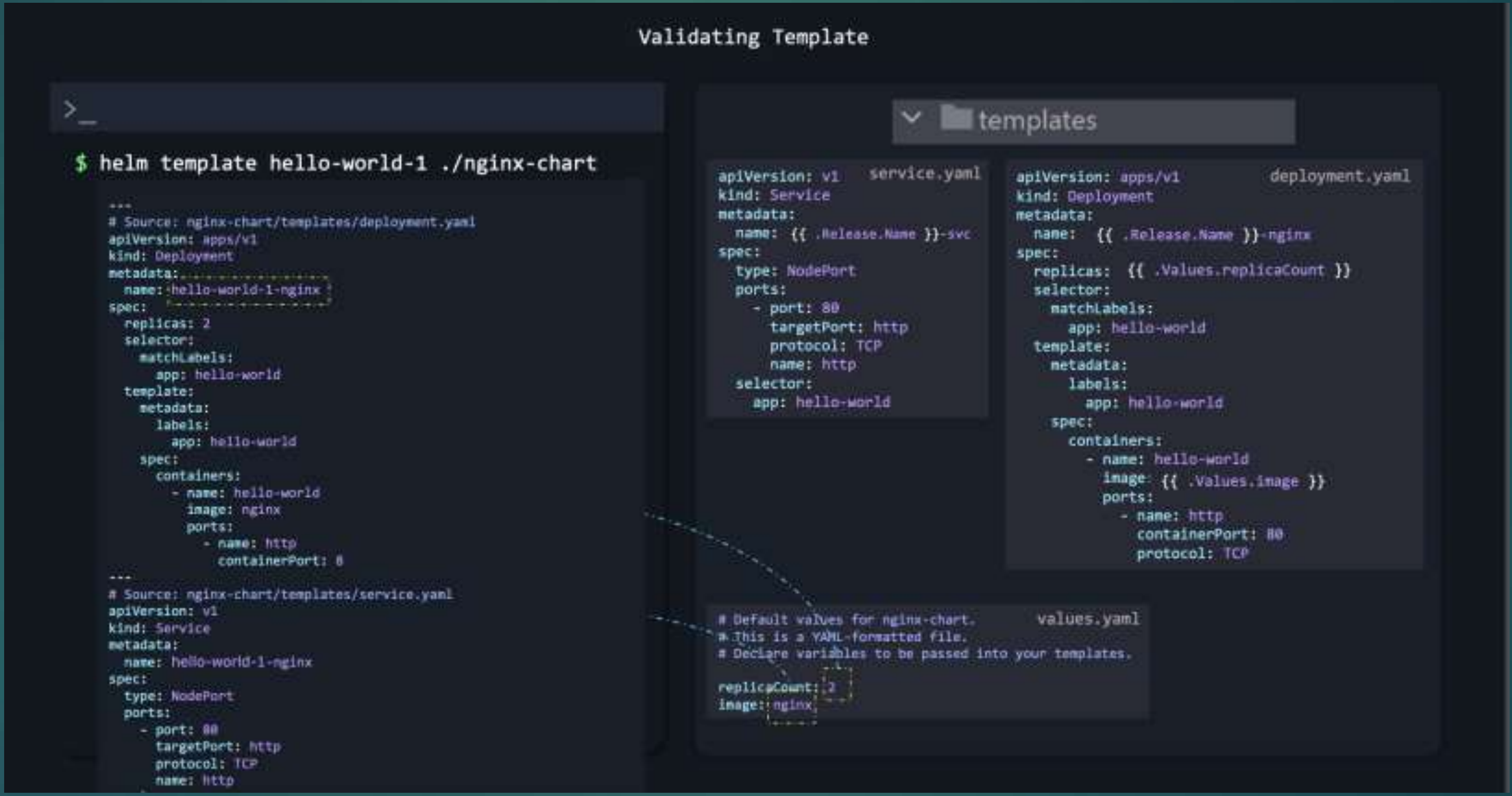
deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ .Values.image }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

values.yaml

```
# Default values for nginx-chart.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicaCount: 2
image: nginx
```

2-templete



Validating Template

```
$ helm template hello-world-1 ./nginx-chart
---
# Source: nginx-chart/templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world-1-nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: nginx
          ports:
            - name: http
              containerPort: 8
---
# Source: nginx-chart/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-world-1-nginx
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
```

service.yaml

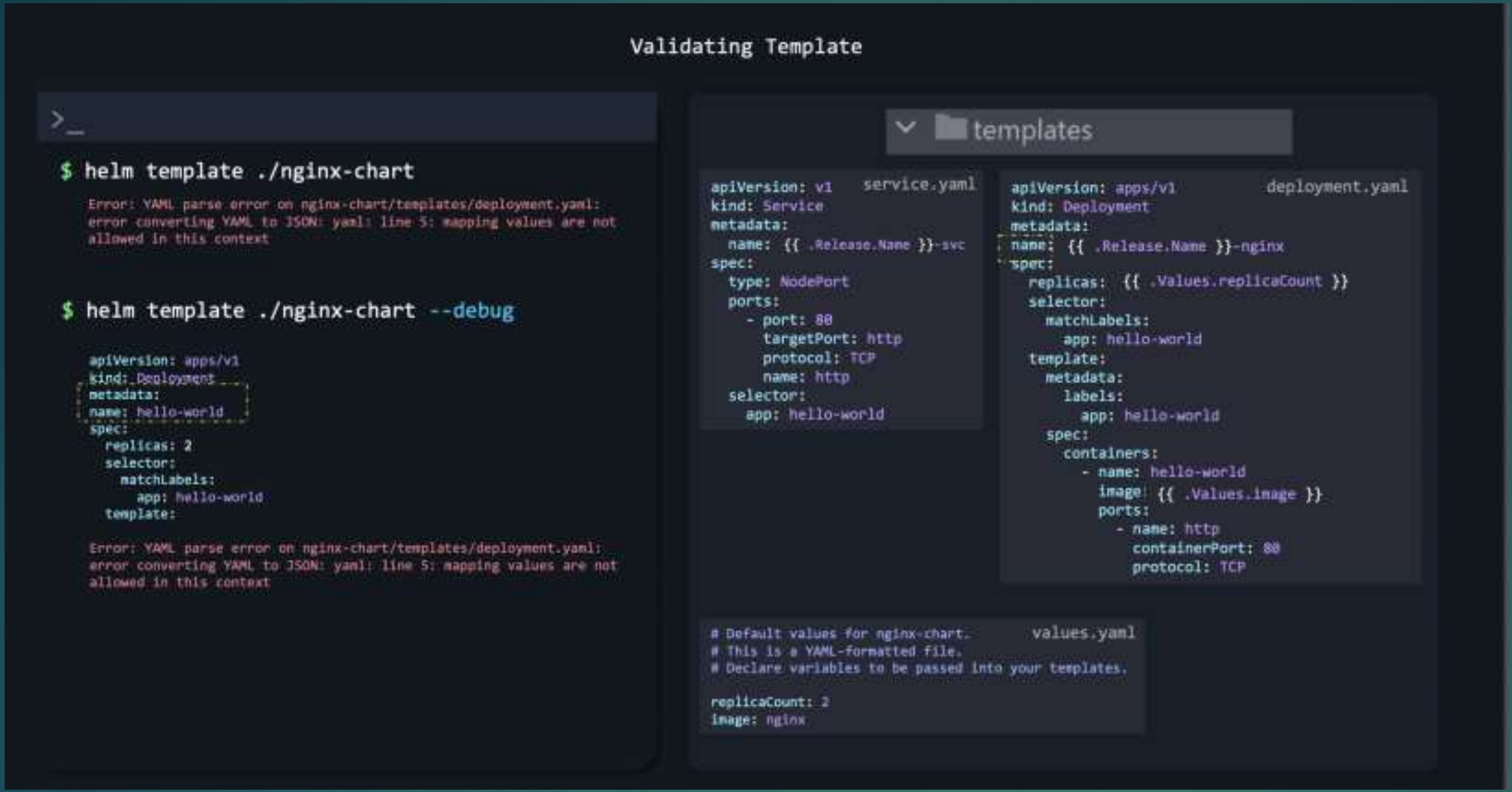
```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-svc
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name:  {{ .Release.Name }}-nginx
spec:
  replicas:  {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ .Values.image }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

values.yaml

```
# Default values for nginx-chart.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicaCount: 2
image: nginx
```



Validating Template

```
>_

$ helm template ./nginx-chart

    Error: YAML parse error on nginx-chart/templates/deployment.yaml:
    error converting YAML to JSON: yaml: line 5: mapping values are not
    allowed in this context


$ helm template ./nginx-chart --debug

    apiVersion: apps/v1
    kind: Deployment
    metadata:
    name: hello-world
    spec:
      replicas: 2
      selector:
        matchLabels:
          app: hello-world
      template:

    Error: YAML parse error on nginx-chart/templates/deployment.yaml:
    error converting YAML to JSON: yaml: line 5: mapping values are not
    allowed in this context
```

templates

service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-svc
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

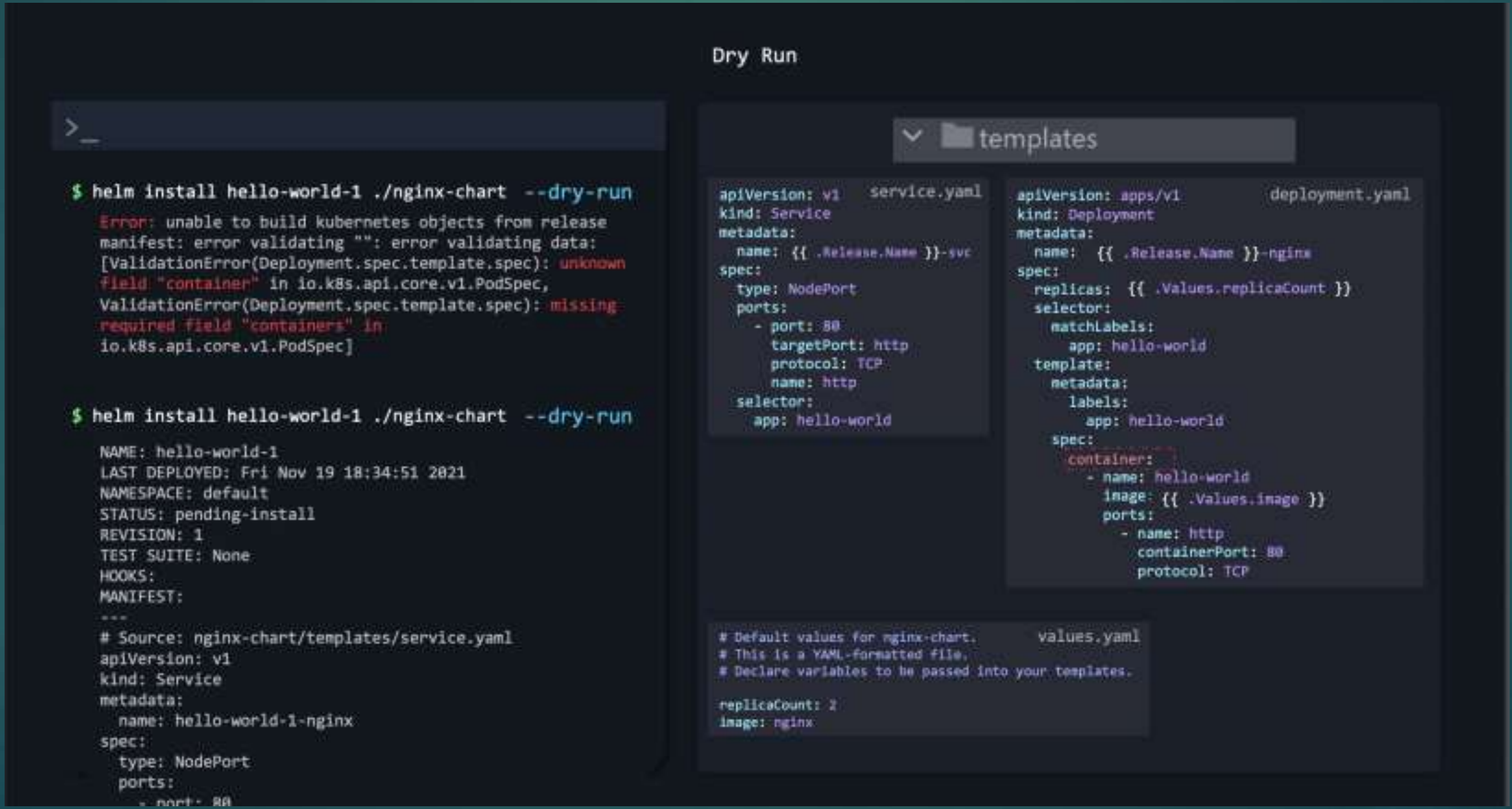
deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: {{ .Release.Name }}-nginx
spec:
  replicas:  {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ .Values.image }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

values.yaml

```
# Default values for nginx-chart.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicaCount: 2
image: nginx
```

3-dry-run



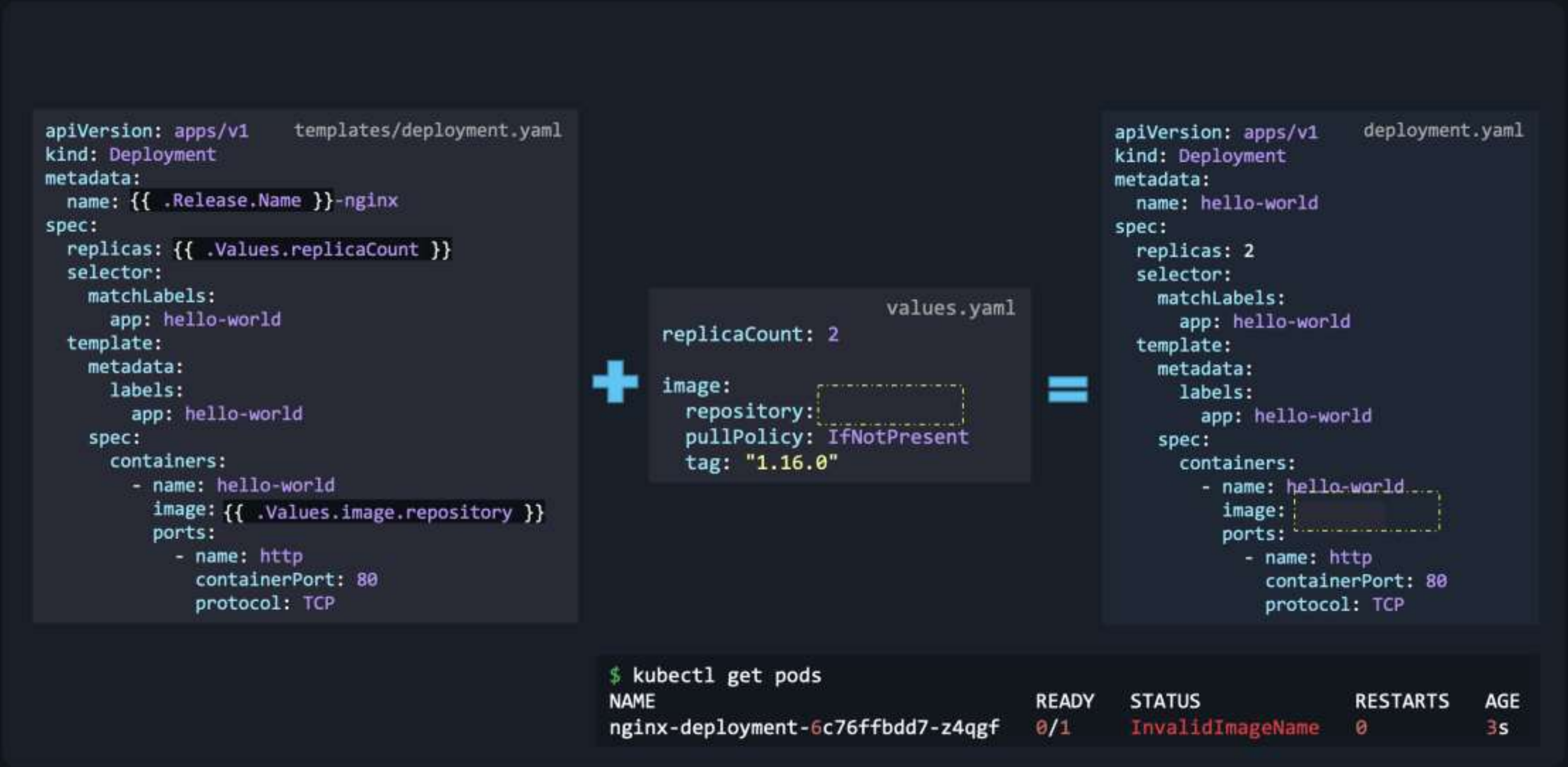
```
$ helm install hello-world-1 ./nginx-chart  --dry-run
  Error: unable to build kubernetes objects from release
  manifest: error validating "": error validating data:
  [ValidationError(Deployment.spec.template.spec): unknown
  field "container" in io.k8s.api.core.v1.PodSpec,
  ValidationError(Deployment.spec.template.spec): missing
  required field "containers" in
  io.k8s.api.core.v1.PodSpec]


$ helm install hello-world-1 ./nginx-chart  --dry-run
  NAME: hello-world-1
  LAST DEPLOYED: Fri Nov 19 18:34:51 2021
  NAMESPACE: default
  STATUS: pending-install
  REVISION: 1
  TEST SUITE: None
  HOOKS:
  MANIFEST:
  ---
  # Source: nginx-chart/templates/service.yaml
  apiVersion: v1
  kind: Service
  metadata:
    name: hello-world-1-nginx
  spec:
    type: NodePort
    ports:
      - port: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-svc
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name:  {{ .Release.Name }}-nginx
spec:
  replicas:  {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      container:
        - name: hello-world
          image: {{ .Values.image }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

```
# Default values for nginx-chart.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicaCount: 2
image: nginx
```

Functions



templates/deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ .Values.image.repository }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

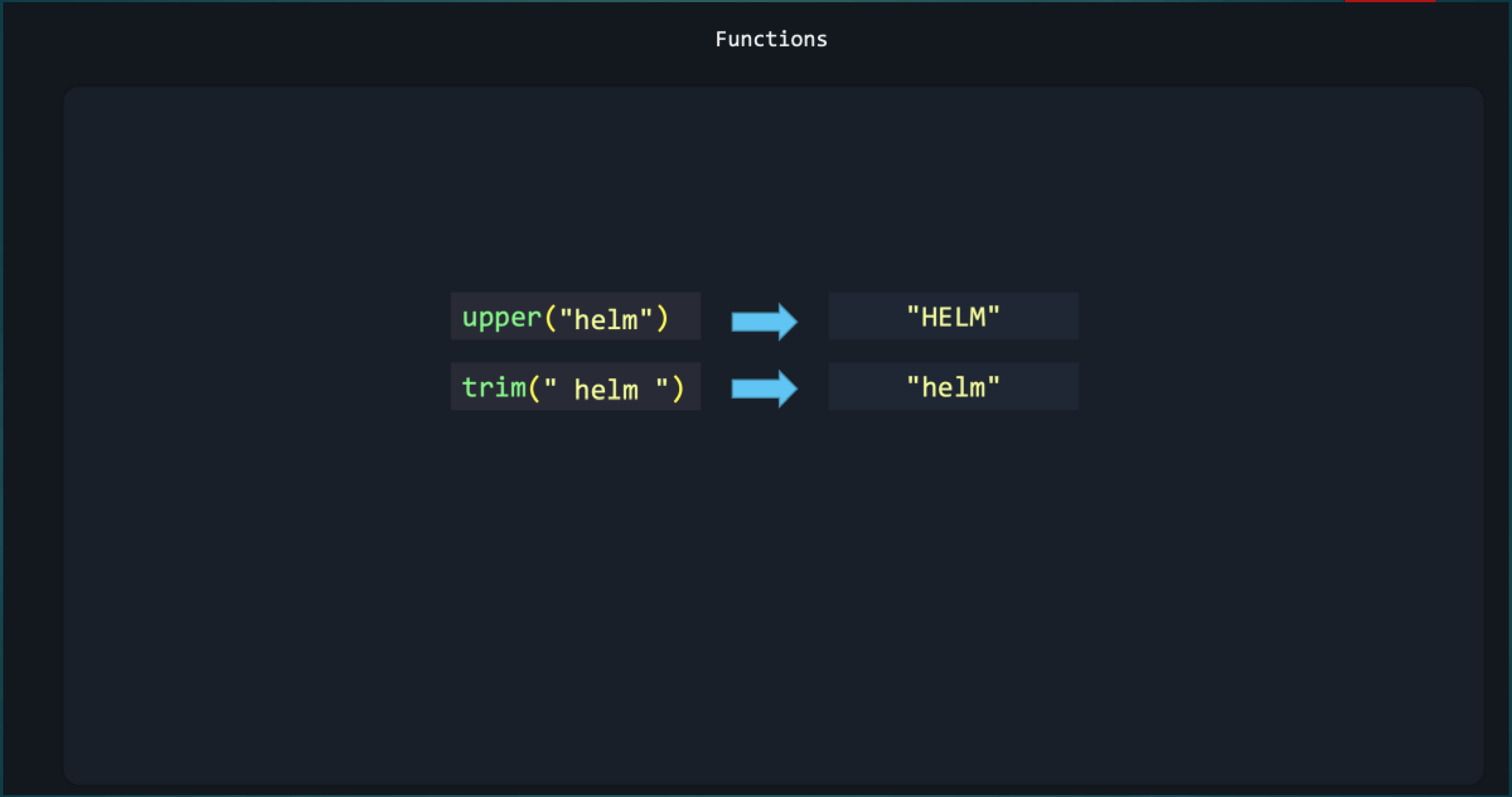
values.yaml

```yaml
replicaCount: 2

image:
  repository:
  pullPolicy: IfNotPresent
  tag: "1.16.0"
```

deployment.yaml

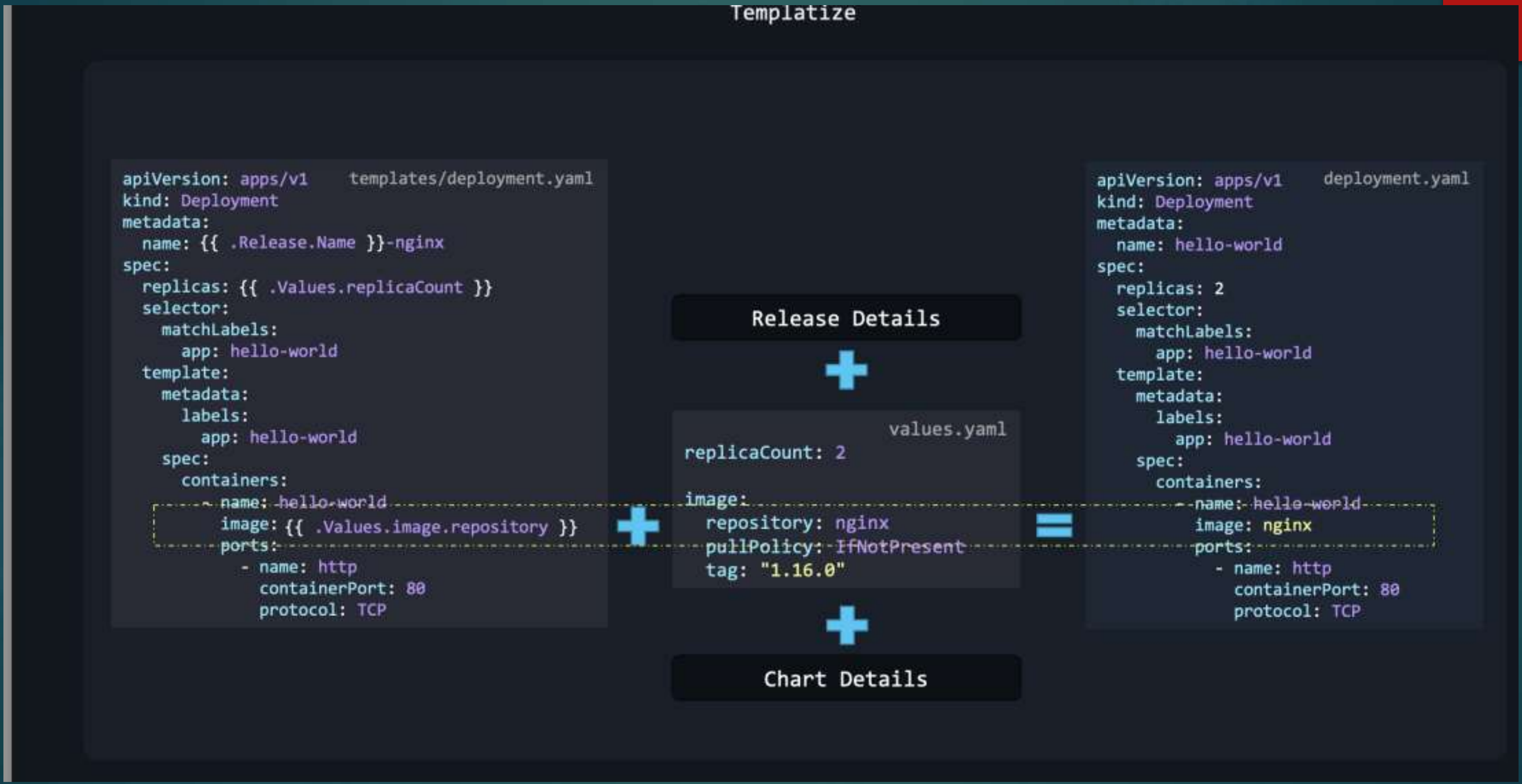
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image:
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

```
$ kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
nginx-deployment-6c76ffbdd7-z4qgf   0/1     InvalidImageName   0          3s
```



Functions

```
upper("helm")     →     "HELM"
trim(" helm ")    →     "helm"
```



Templatize

templates/deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
      - name: hello-world
        image: {{ .Values.image.repository }}
        ports:
          - name: http
            containerPort: 80
            protocol: TCP
```

Release Details

+

values.yaml

```yaml
replicaCount: 2

image:
  repository: nginx
  pullPolicy: IfNotPresent
  tag: "1.16.0"
```

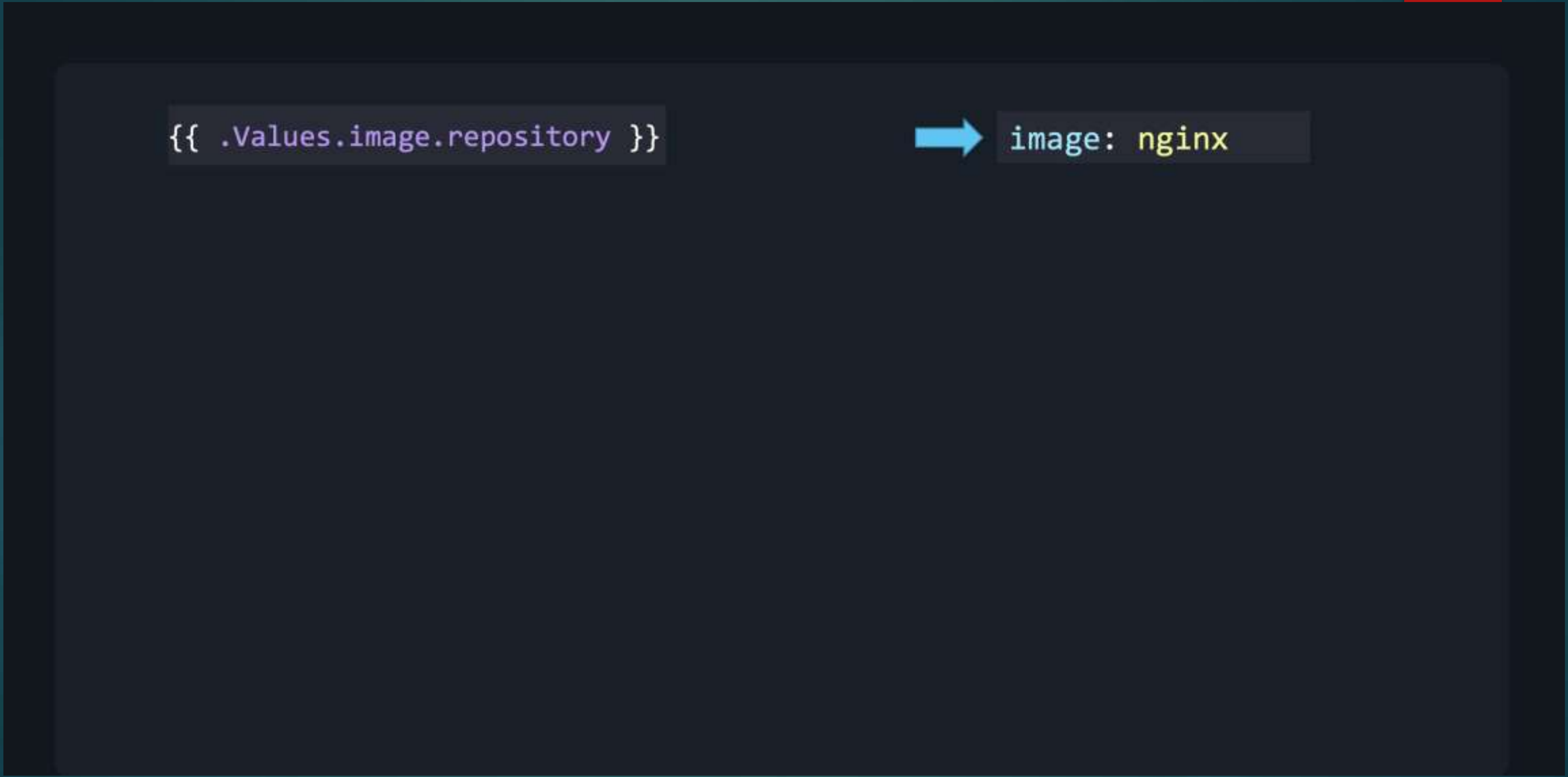
+

Chart Details

=

deployment.yaml

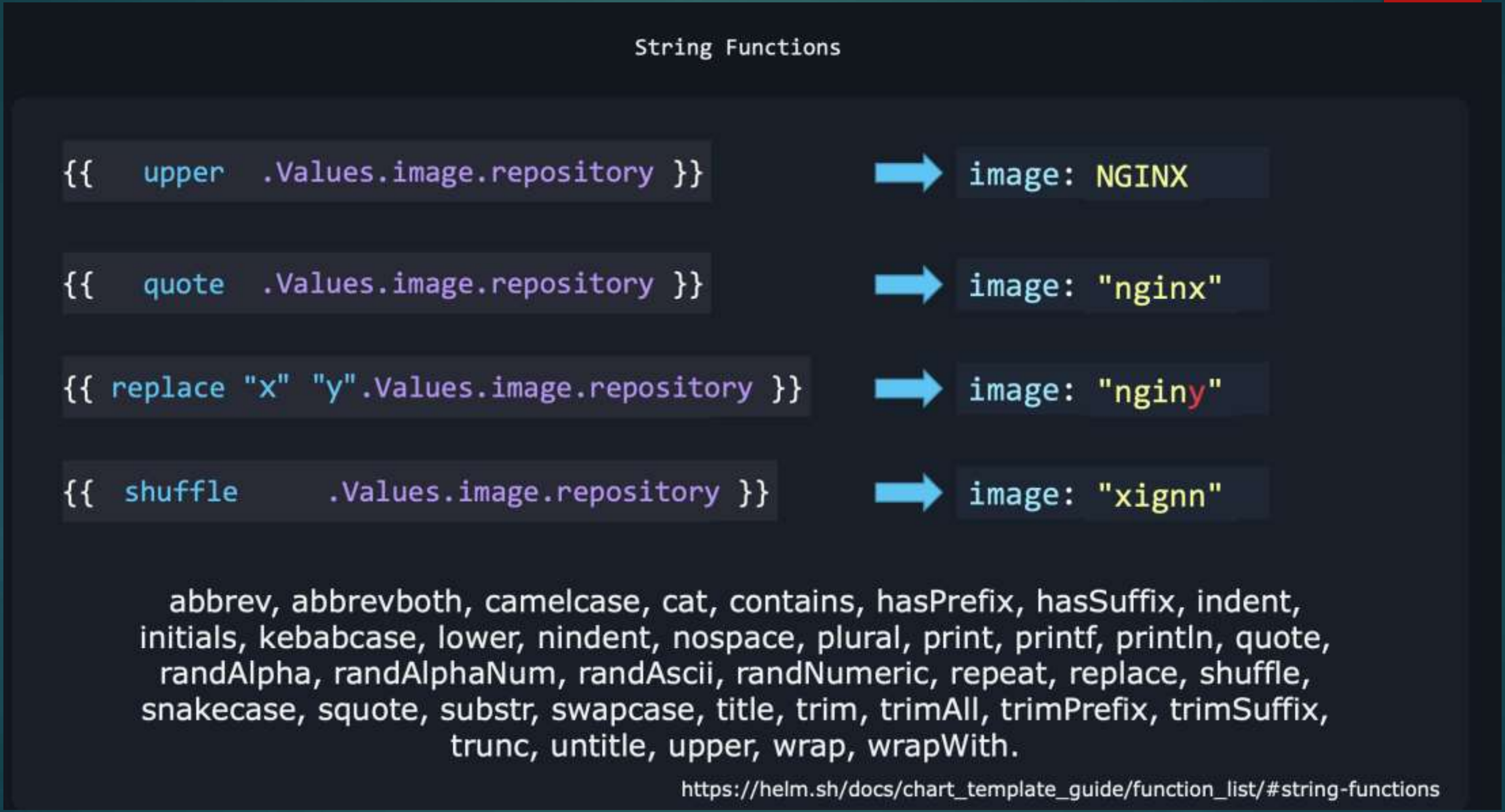
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
      - name: hello-world
        image: nginx
        ports:
          - name: http
            containerPort: 80
            protocol: TCP
```



```
{{ .Values.image.repository }}
```

→

```
image: nginx
```

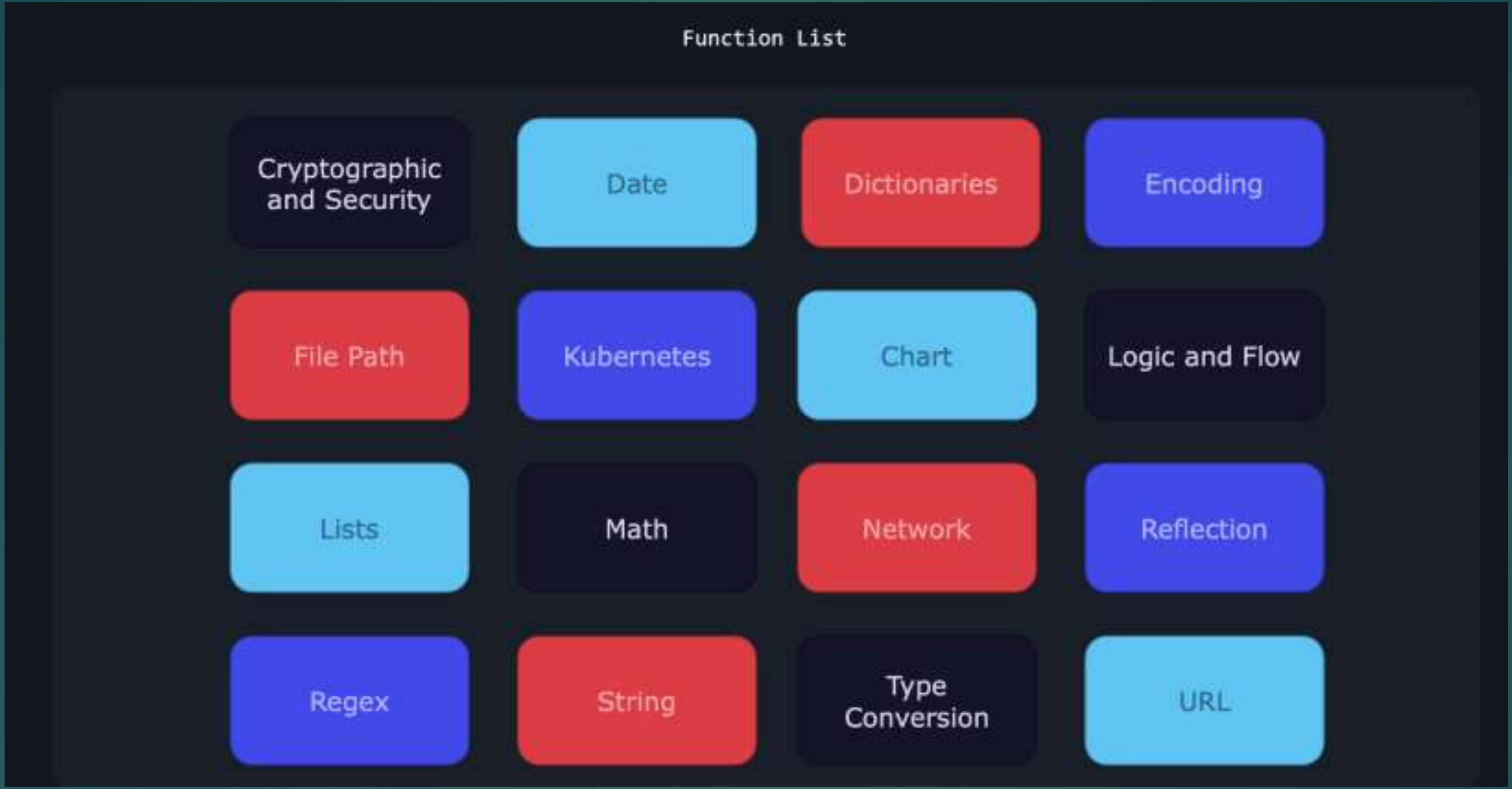


String Functions

```
{{   upper   .Values.image.repository }}        →   image: NGINX

{{   quote   .Values.image.repository }}        →   image: "nginx"

{{ replace "x"  "y".Values.image.repository }}  →   image: "niny"

{{  shuffle      .Values.image.repository }}    →   image: "xignn"
```

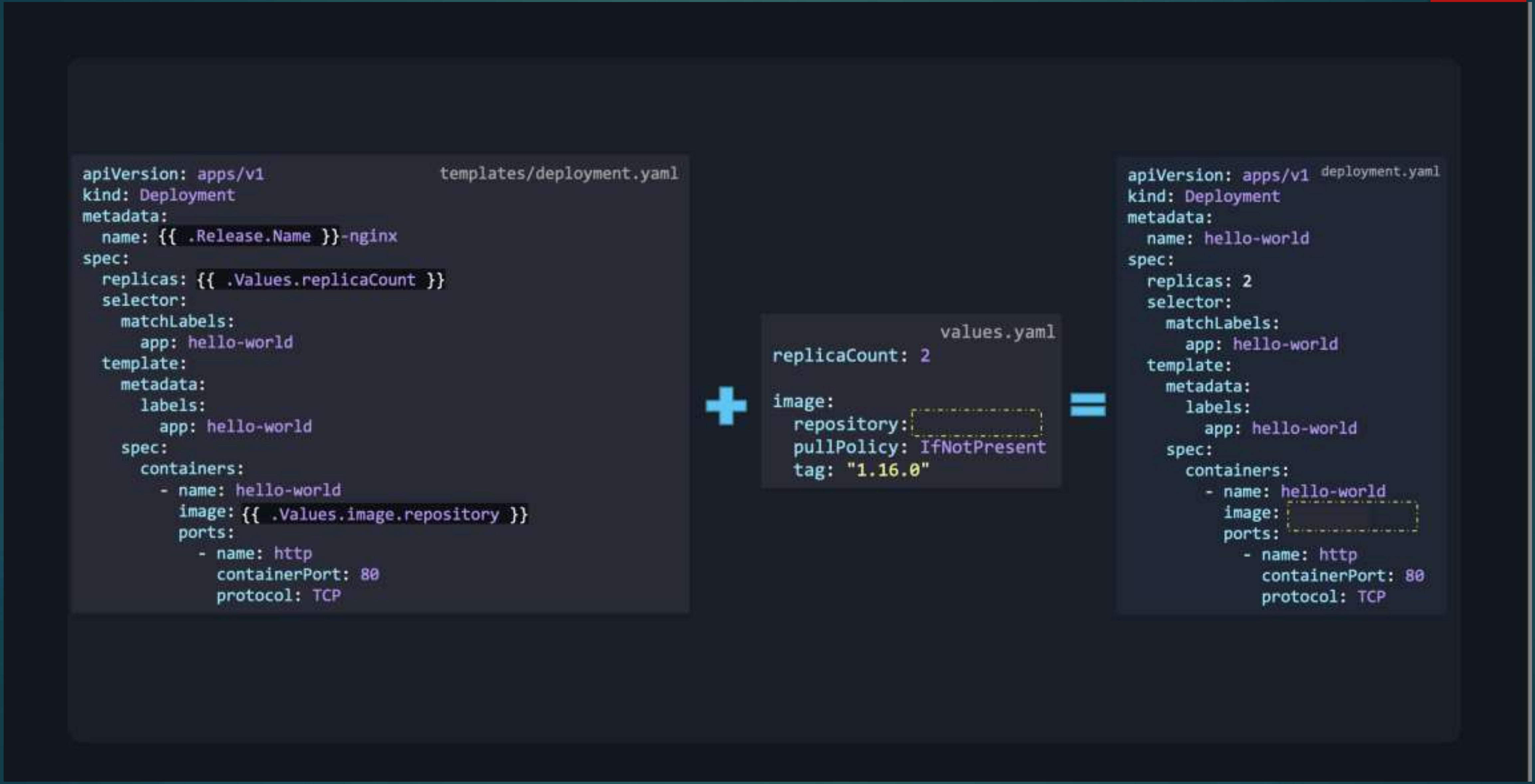
abbrev, abbrevboth, camelcase, cat, contains, hasPrefix, hasSuffix, indent, initials, kebabcase, lower, nindent, nospace, plural, print, printf, println, quote, randAlpha, randAlphaNum, randAscii, randNumeric, repeat, replace, shuffle, snakecase, squote, substr, swapcase, title, trim, trimAll, trimPrefix, trimSuffix, trunc, untitle, upper, wrap, wrapWith.

https://helm.sh/docs/chart_template_guide/function_list/#string-functions



Function List

Cryptographic and Security	Date	Dictionaries	Encoding
File Path	Kubernetes	Chart	Logic and Flow
Lists	Math	Network	Reflection
Regex	String	Type Conversion	URL



templates/deployment.yaml

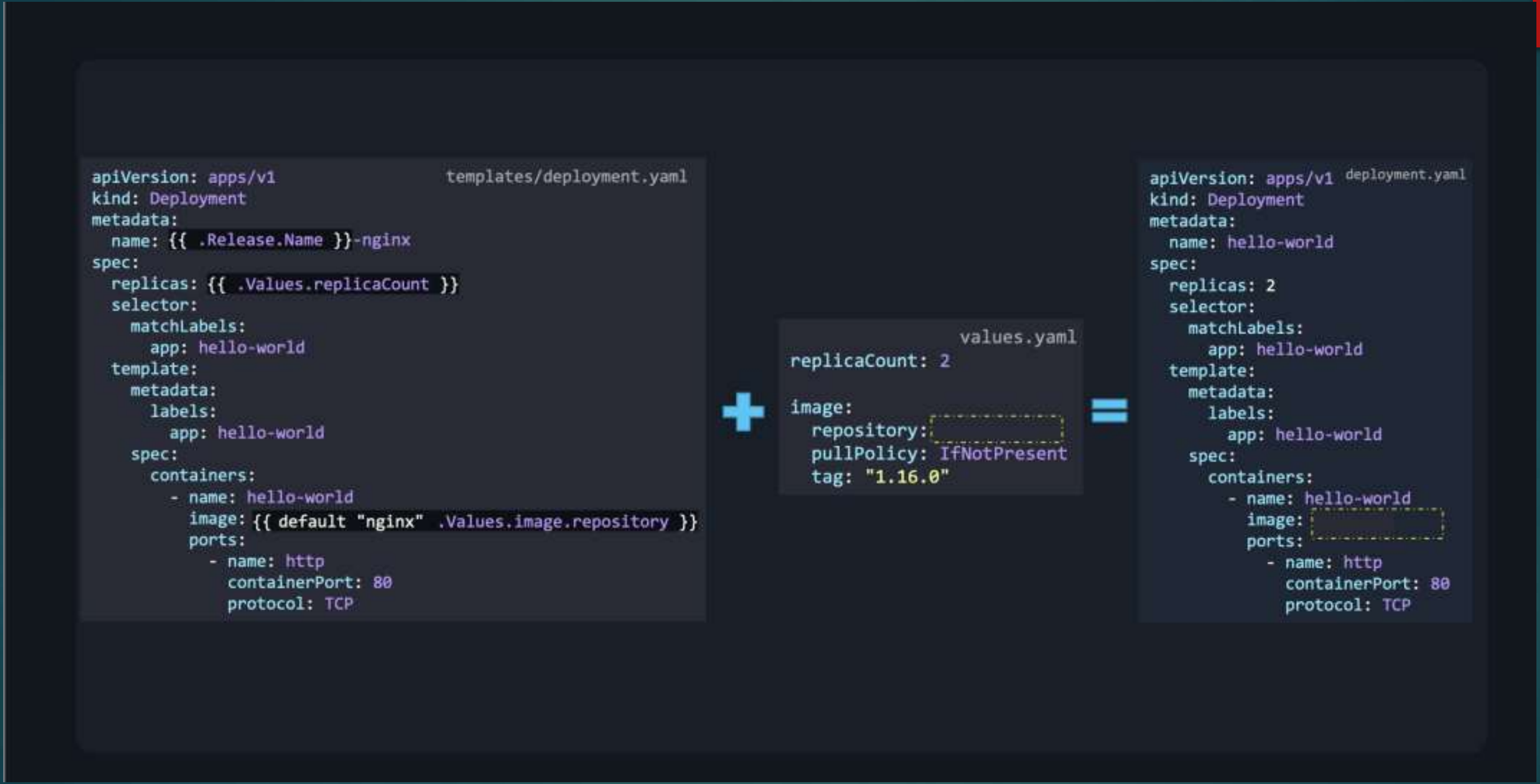
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ .Values.image.repository }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

values.yaml

```yaml
replicaCount: 2

image:
  repository:
  pullPolicy: IfNotPresent
  tag: "1.16.0"
```

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image:
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```



templates/deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: {{ default "nginx" .Values.image.repository }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

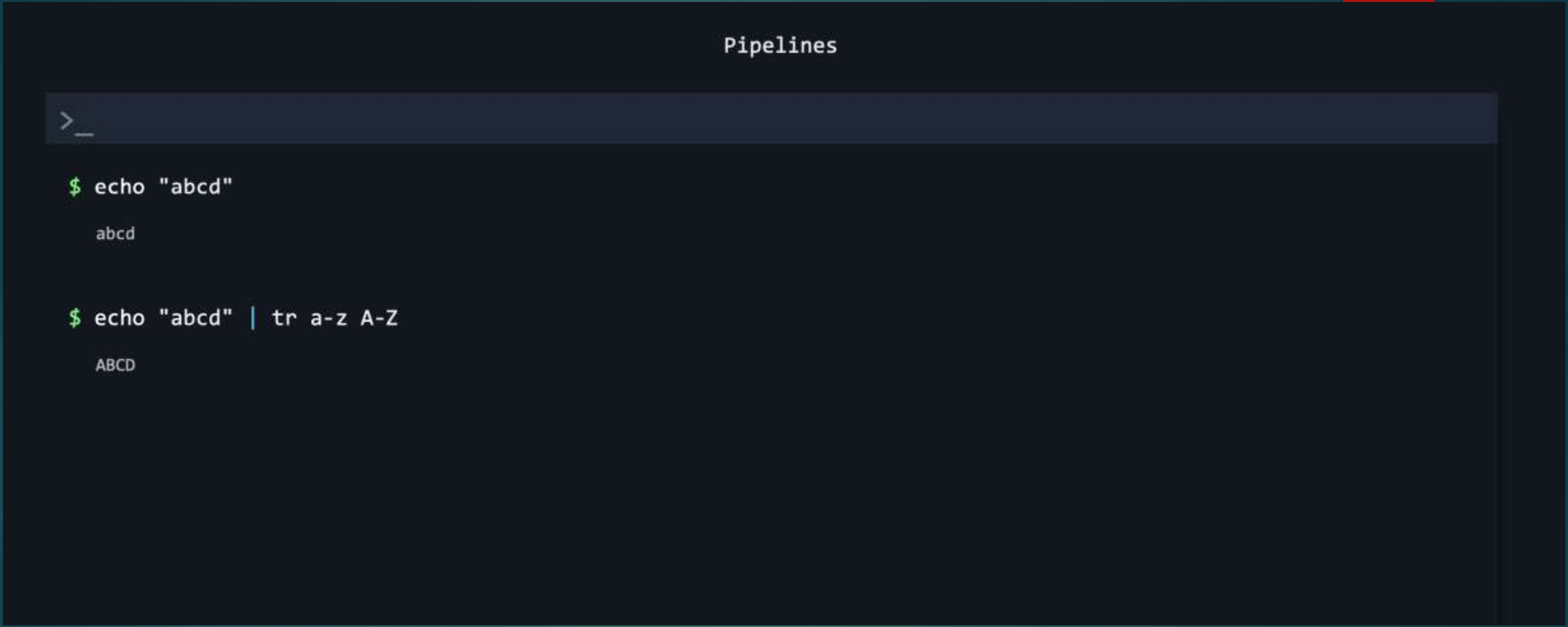
values.yaml

```
replicaCount: 2

image:
  repository:
  pullPolicy: IfNotPresent
  tag: "1.16.0"
```

deployment.yaml

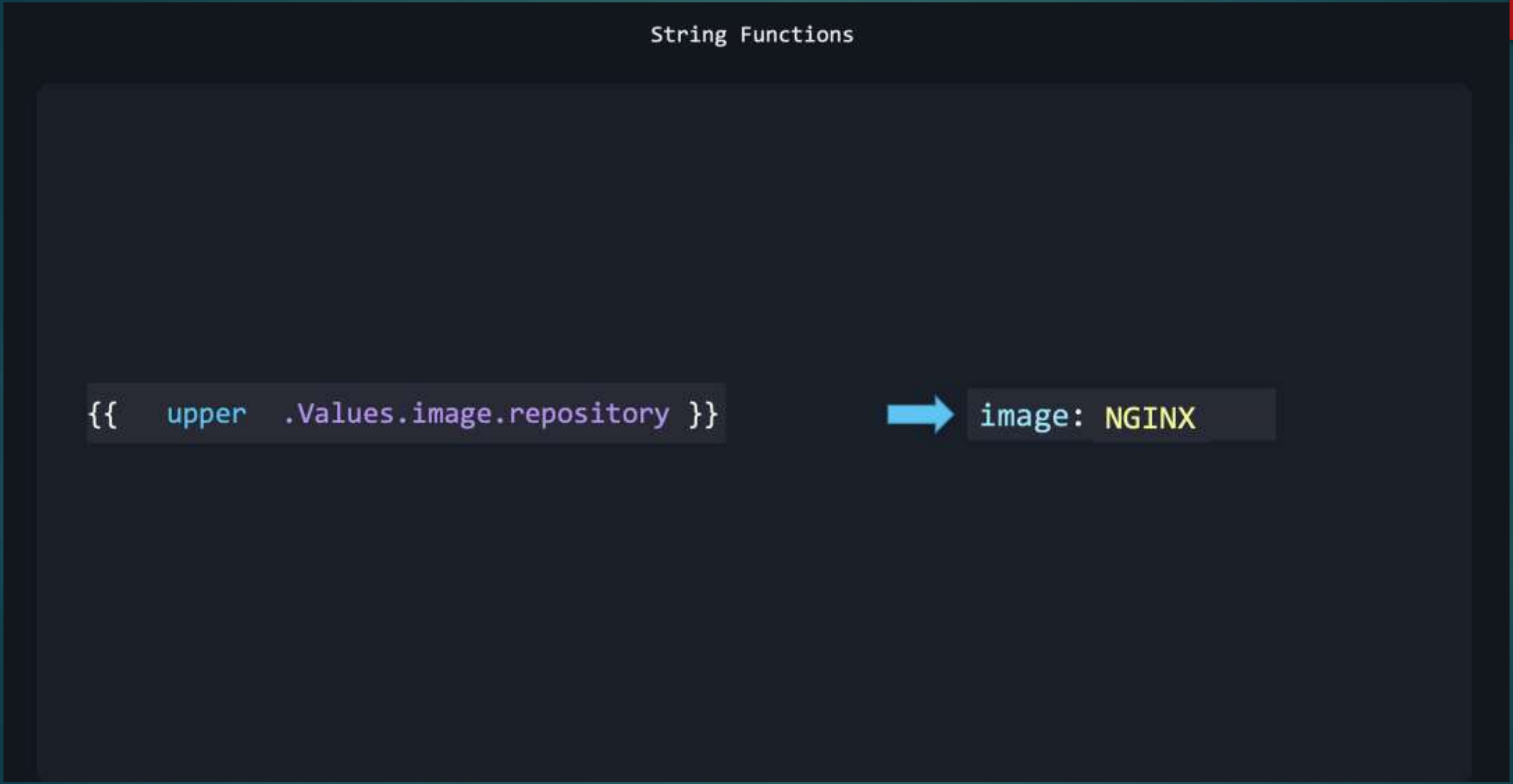
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 2
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image:
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

PipeLines



Pipelines

```
>_

$ echo "abcd"

  abcd


$ echo "abcd" | tr a-z A-Z

  ABCD
```

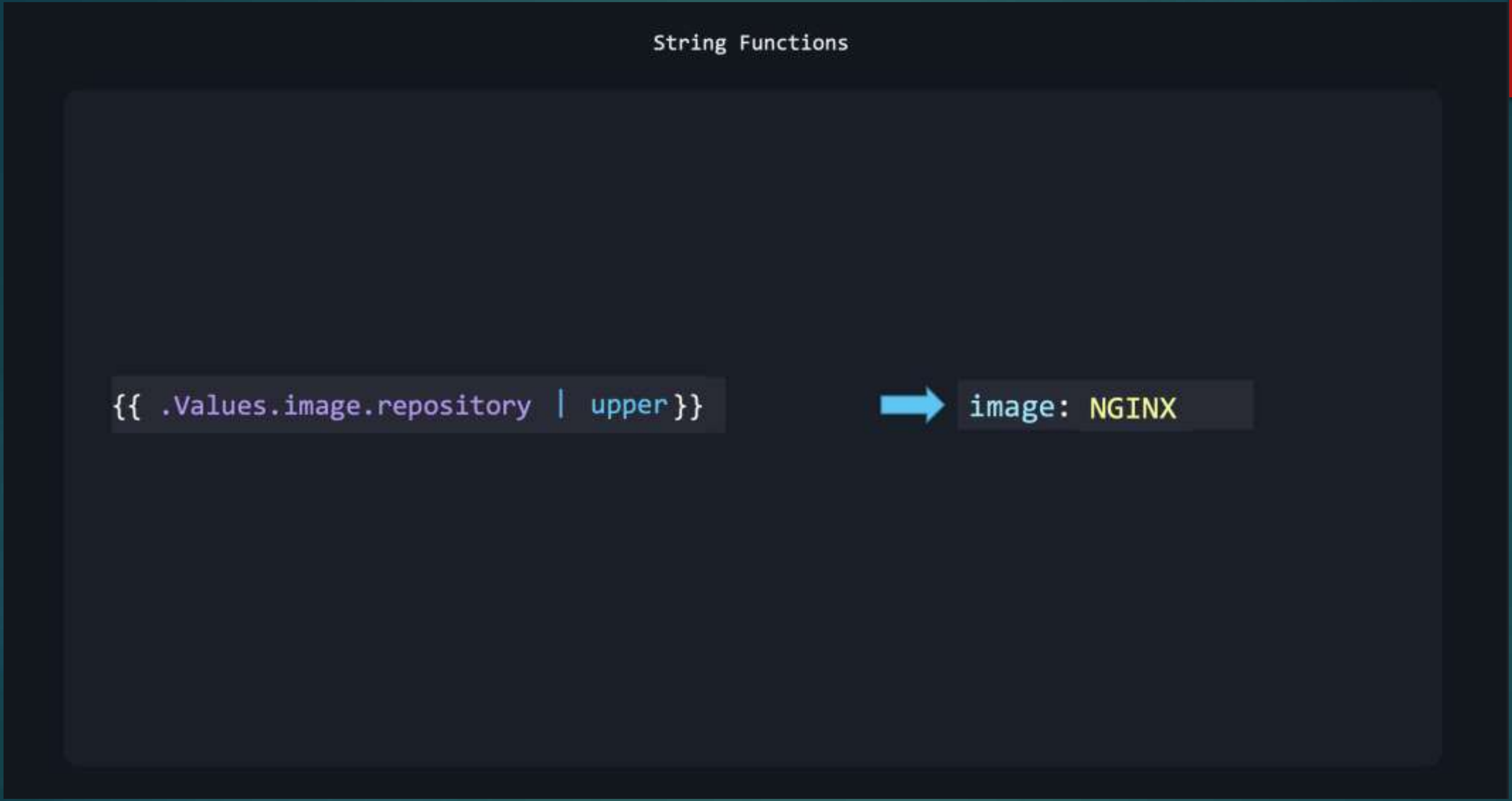


String Functions

```
{{  upper  .Values.image.repository }}
```

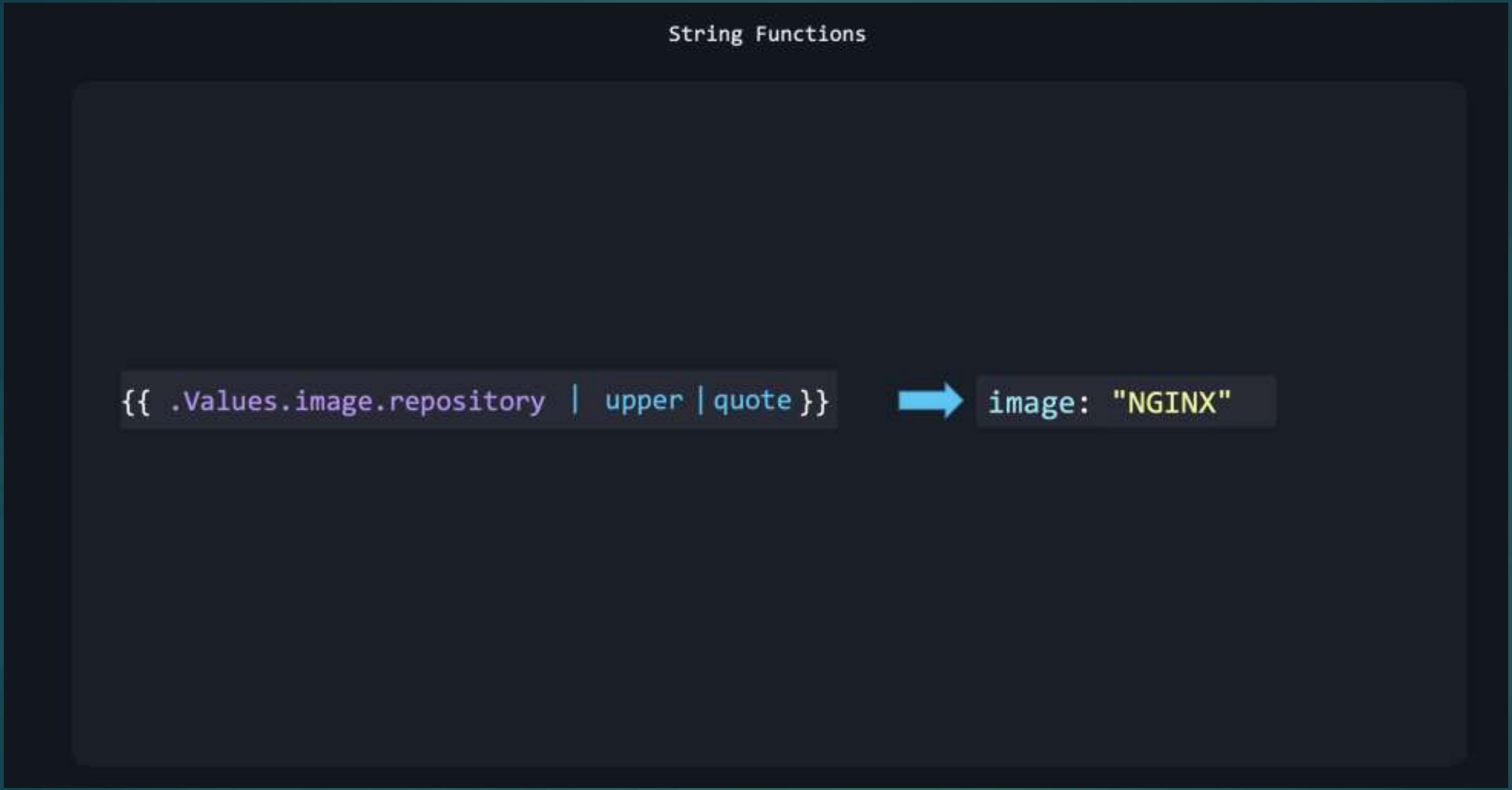
→

```
image: NGINX
```



String Functions

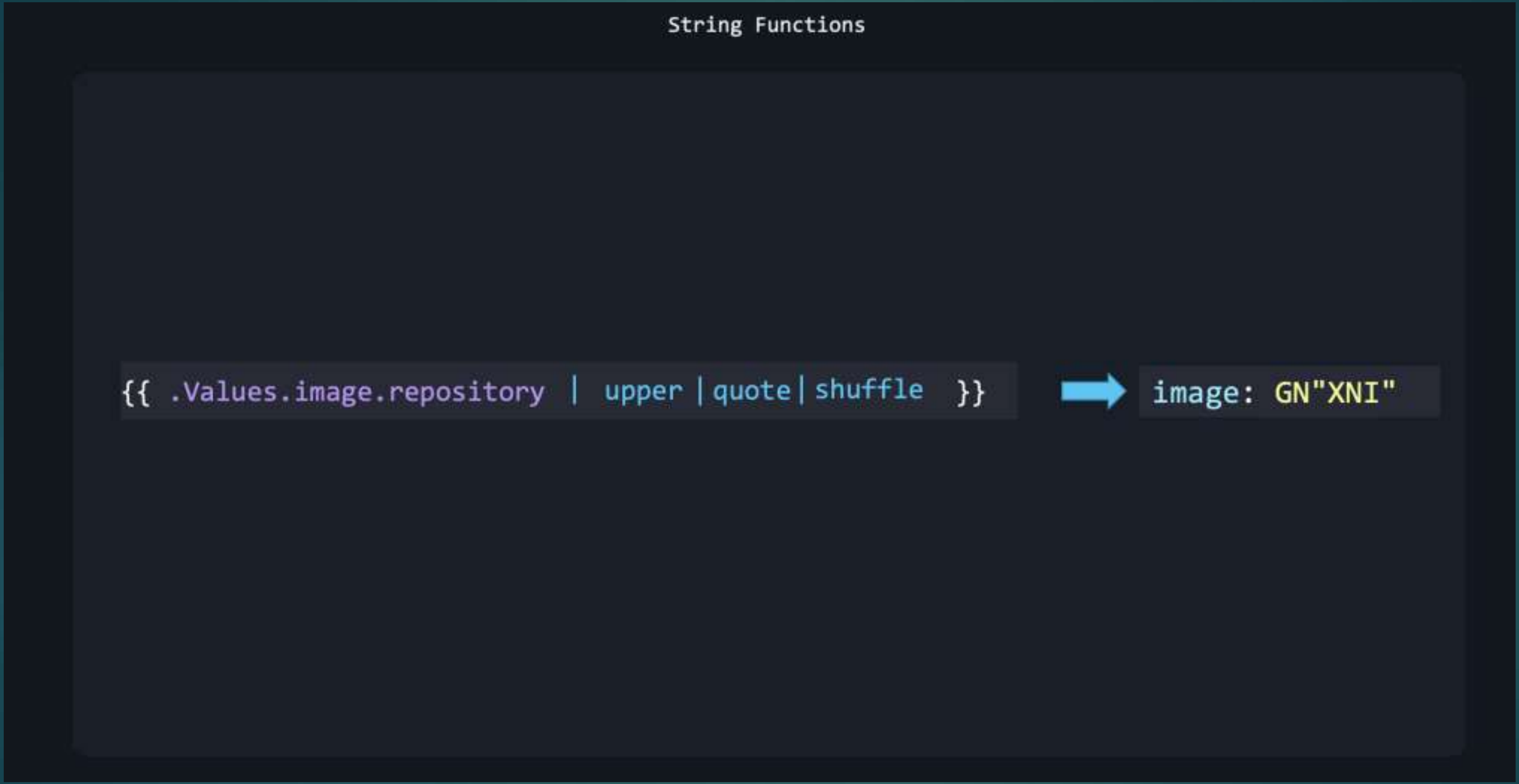
`{{ .Values.image.repository | upper}}` → `image: NGINX`



String Functions

```
{{ .Values.image.repository | upper | quote }}
```

```
image: "NGINX"
```



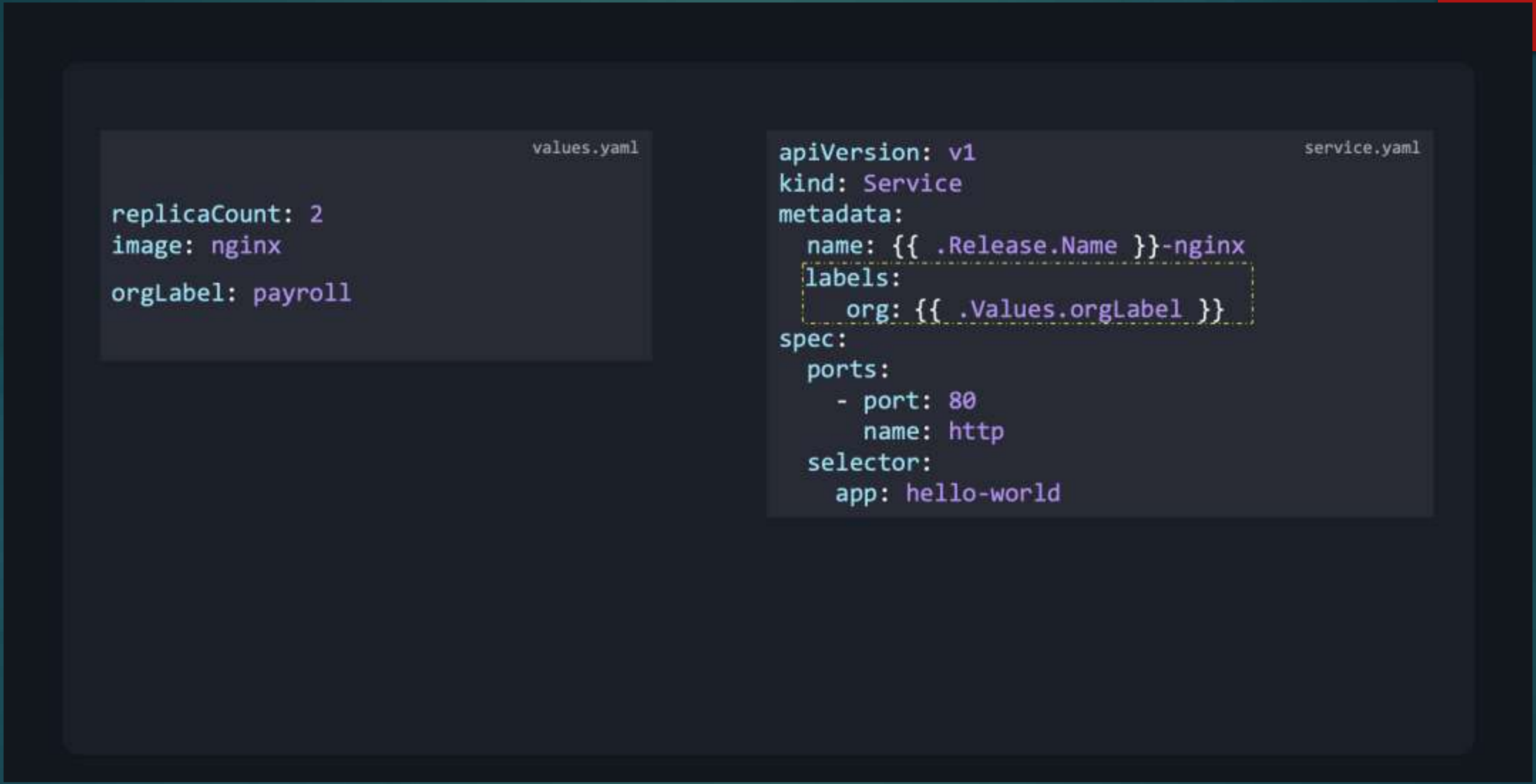
String Functions

```
{{ .Values.image.repository | upper | quote | shuffle }}
```

→

```
image: GN"XNI"
```

Conditionals

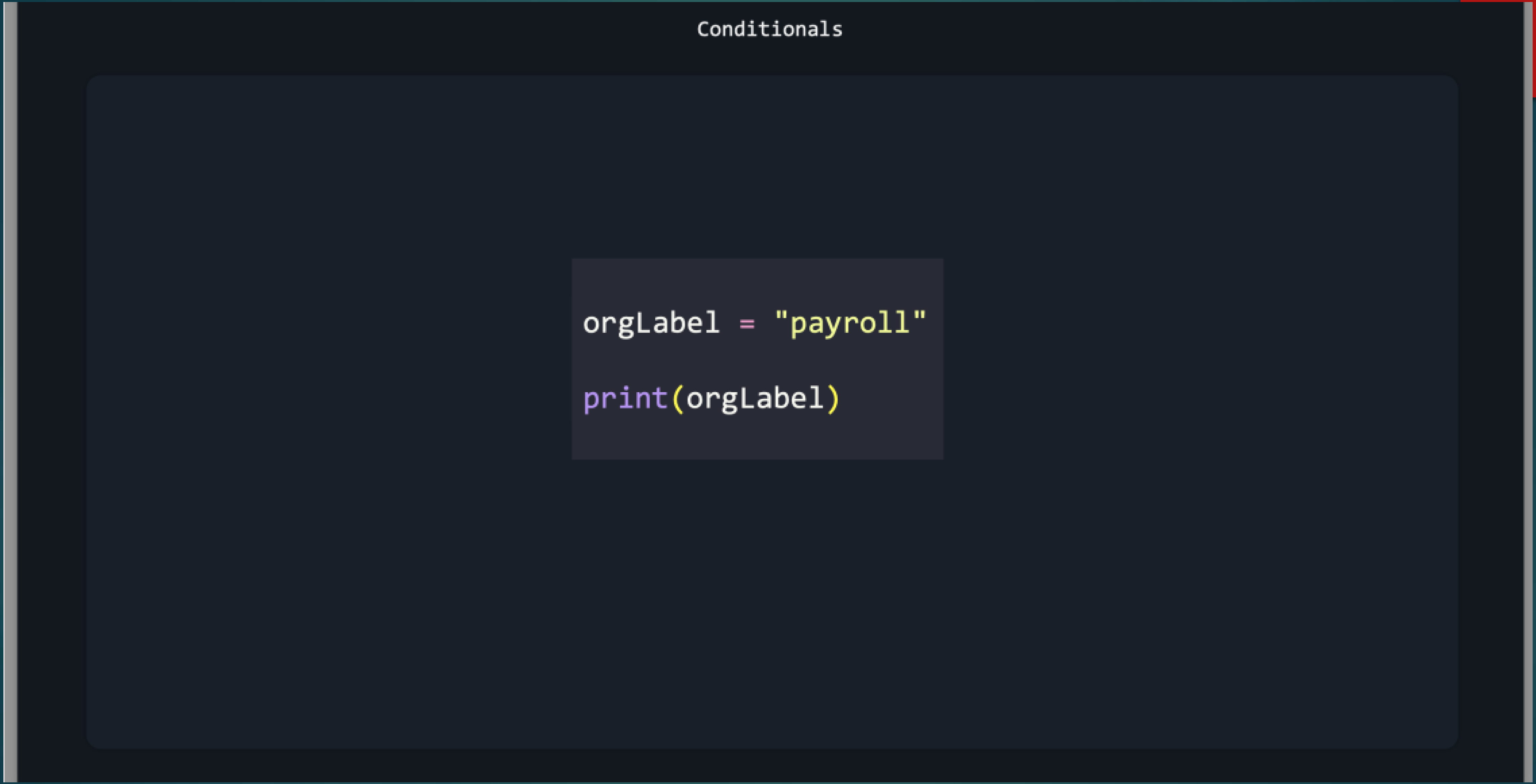


values.yaml

```yaml
replicaCount: 2
image: nginx

orgLabel: payroll
```

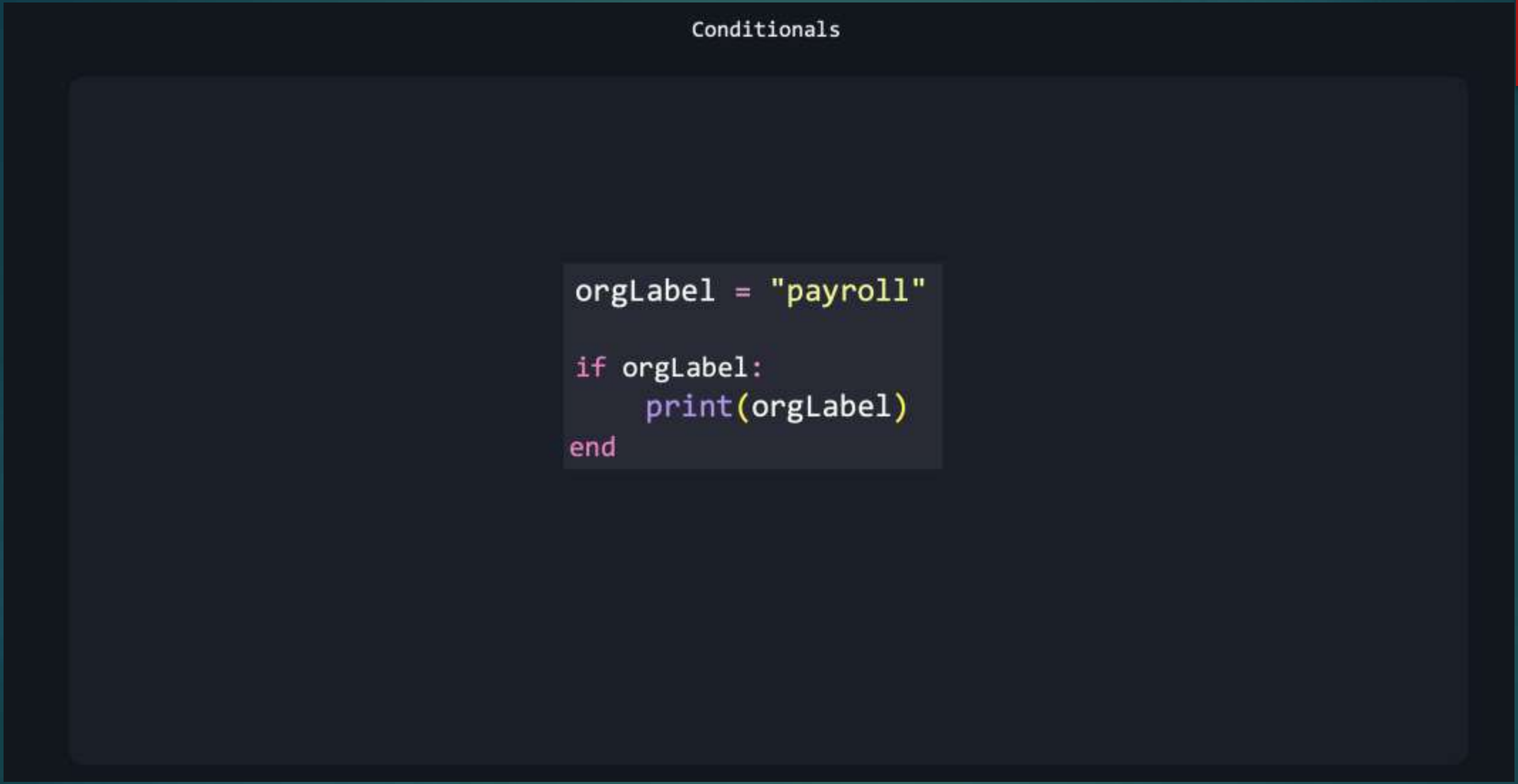
service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-nginx
  labels:
    org: {{ .Values.orgLabel }}
spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```



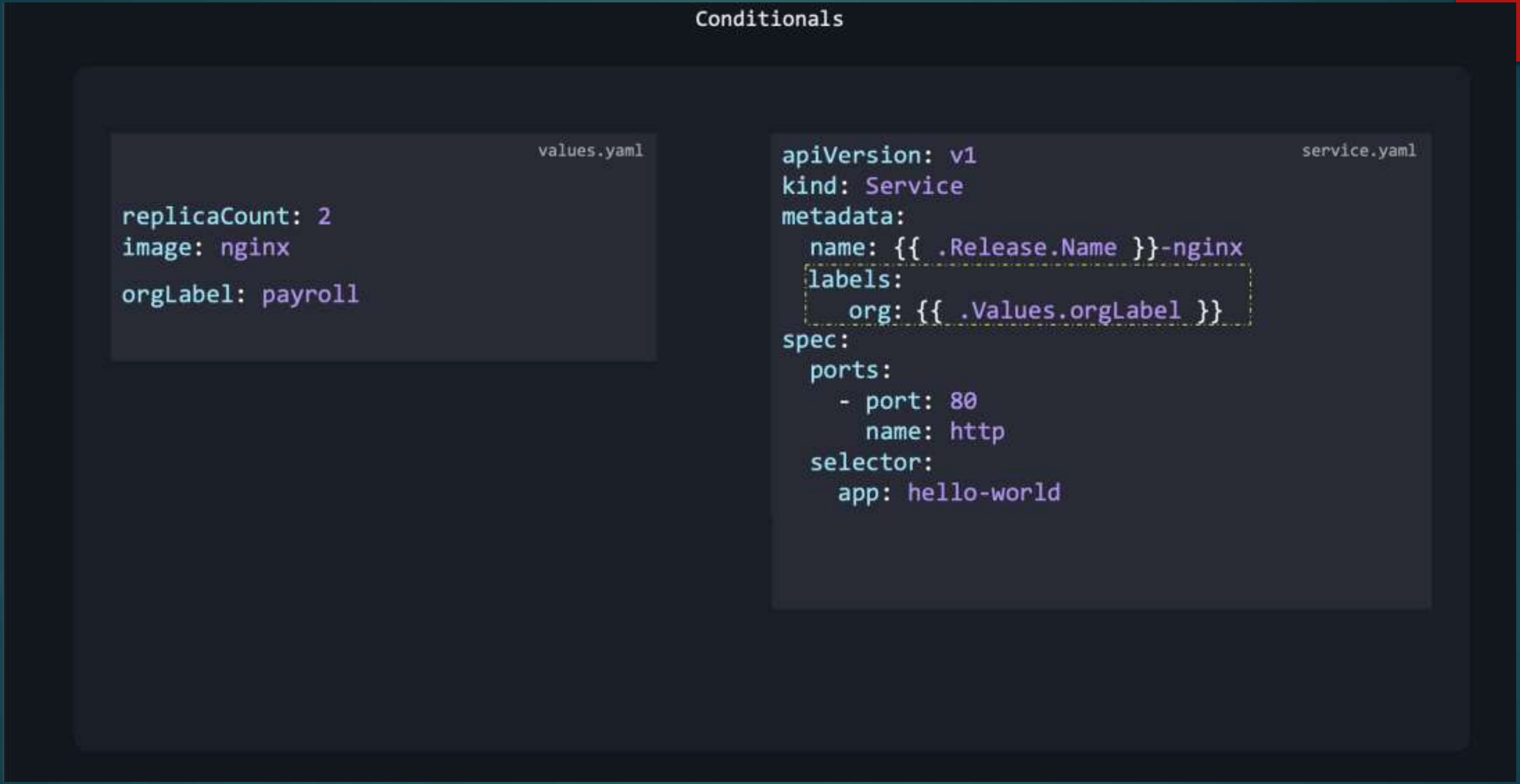
Conditionals

```
orgLabel = "payroll"

print(orgLabel)
```



Conditionals

```
orgLabel = "payroll"

if orgLabel:
    print(orgLabel)
end
```



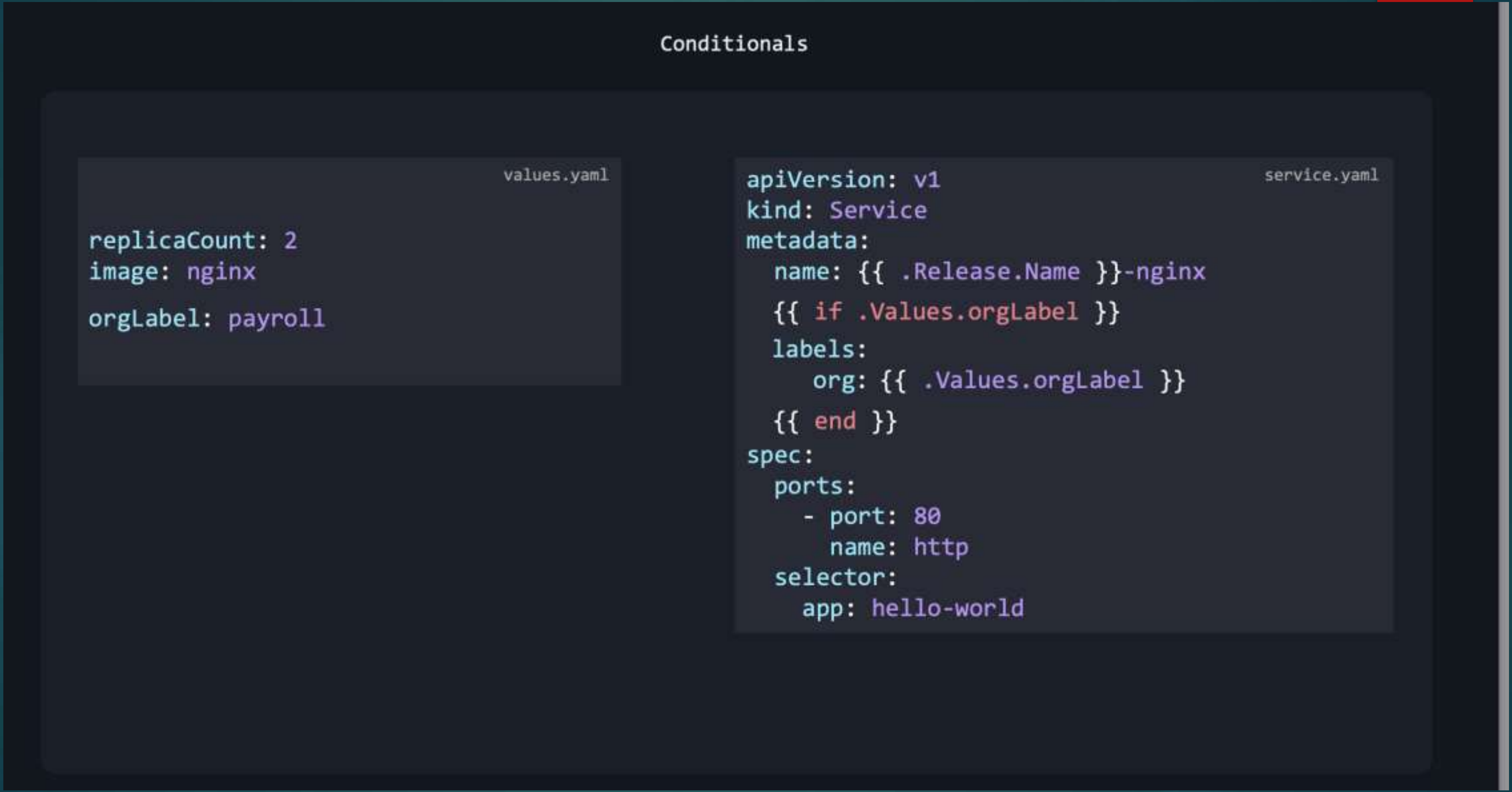
Conditionals

values.yaml

```yaml
replicaCount: 2
image: nginx

orgLabel: payroll
```

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-nginx
  labels:
     org: {{ .Values.orgLabel }}
spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```



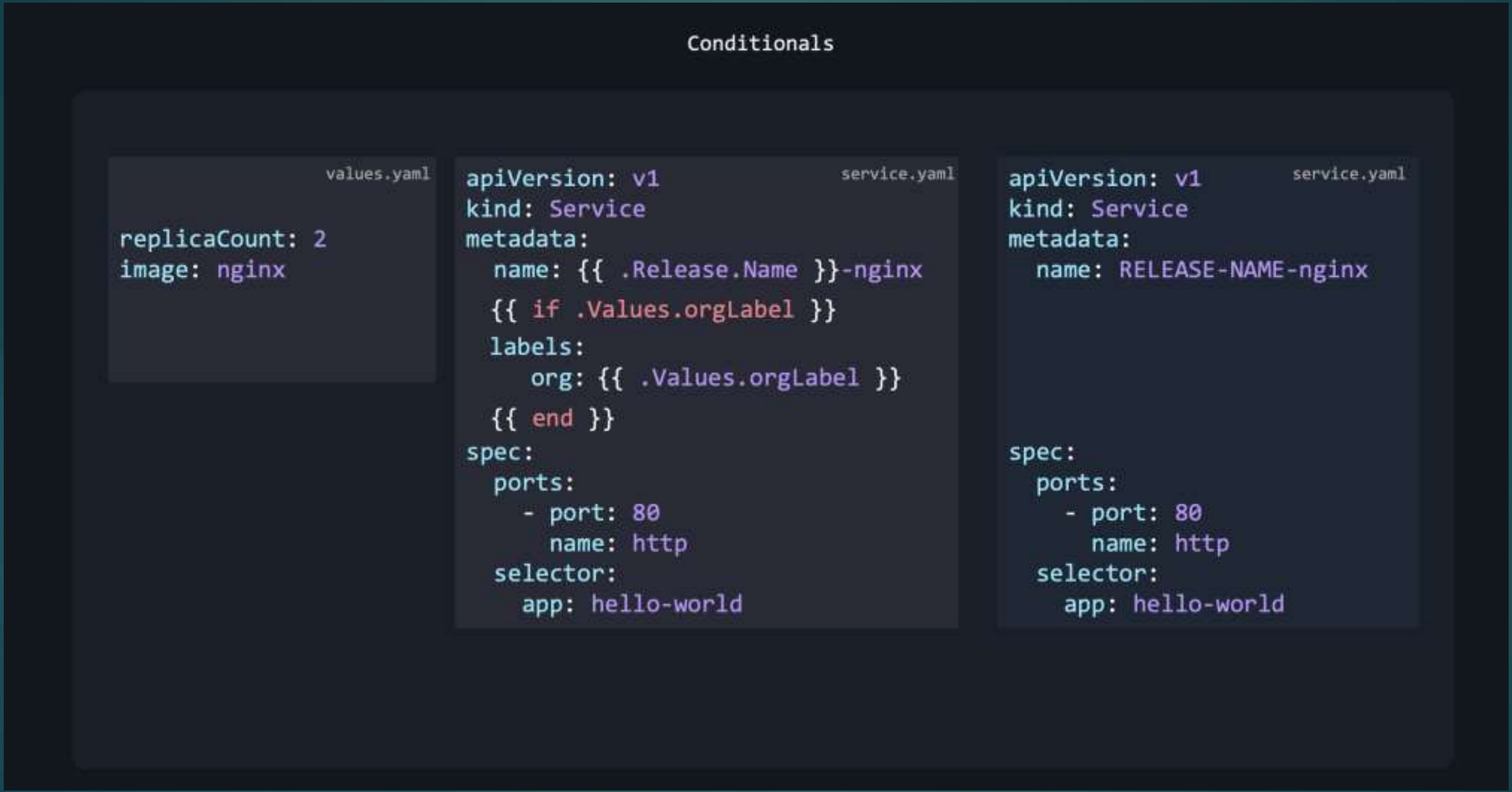
Conditionals

values.yaml

```yaml
replicaCount: 2
image: nginx

orgLabel: payroll
```

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-nginx
  {{ if .Values.orgLabel }}
  labels:
     org: {{ .Values.orgLabel }}
  {{ end }}
spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```



Conditionals

values.yaml

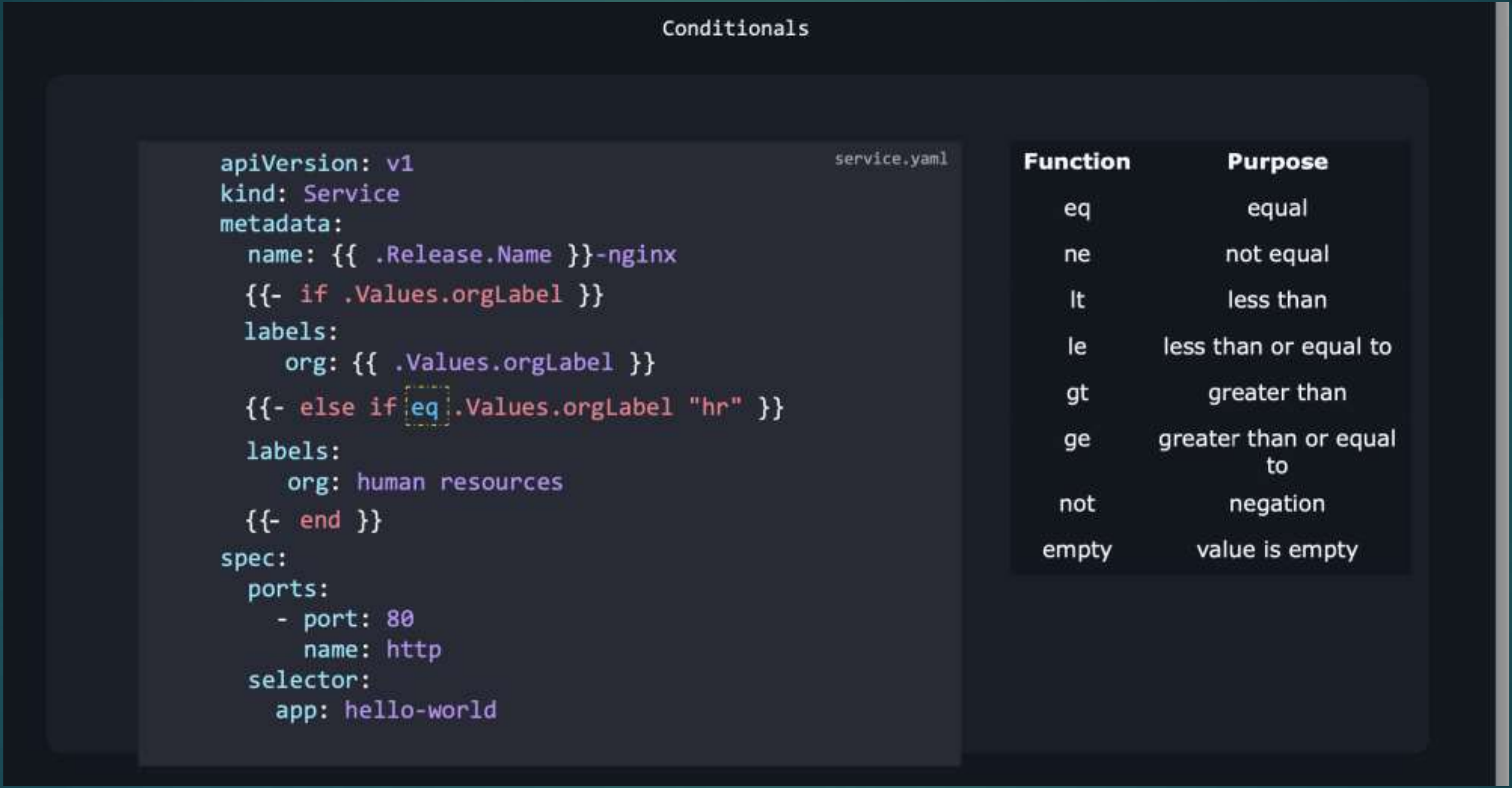
```yaml
replicaCount: 2
image: nginx
```

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-nginx
  {{ if .Values.orgLabel }}
  labels:
    org: {{ .Values.orgLabel }}
  {{ end }}
spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: RELEASE-NAME-nginx
spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```

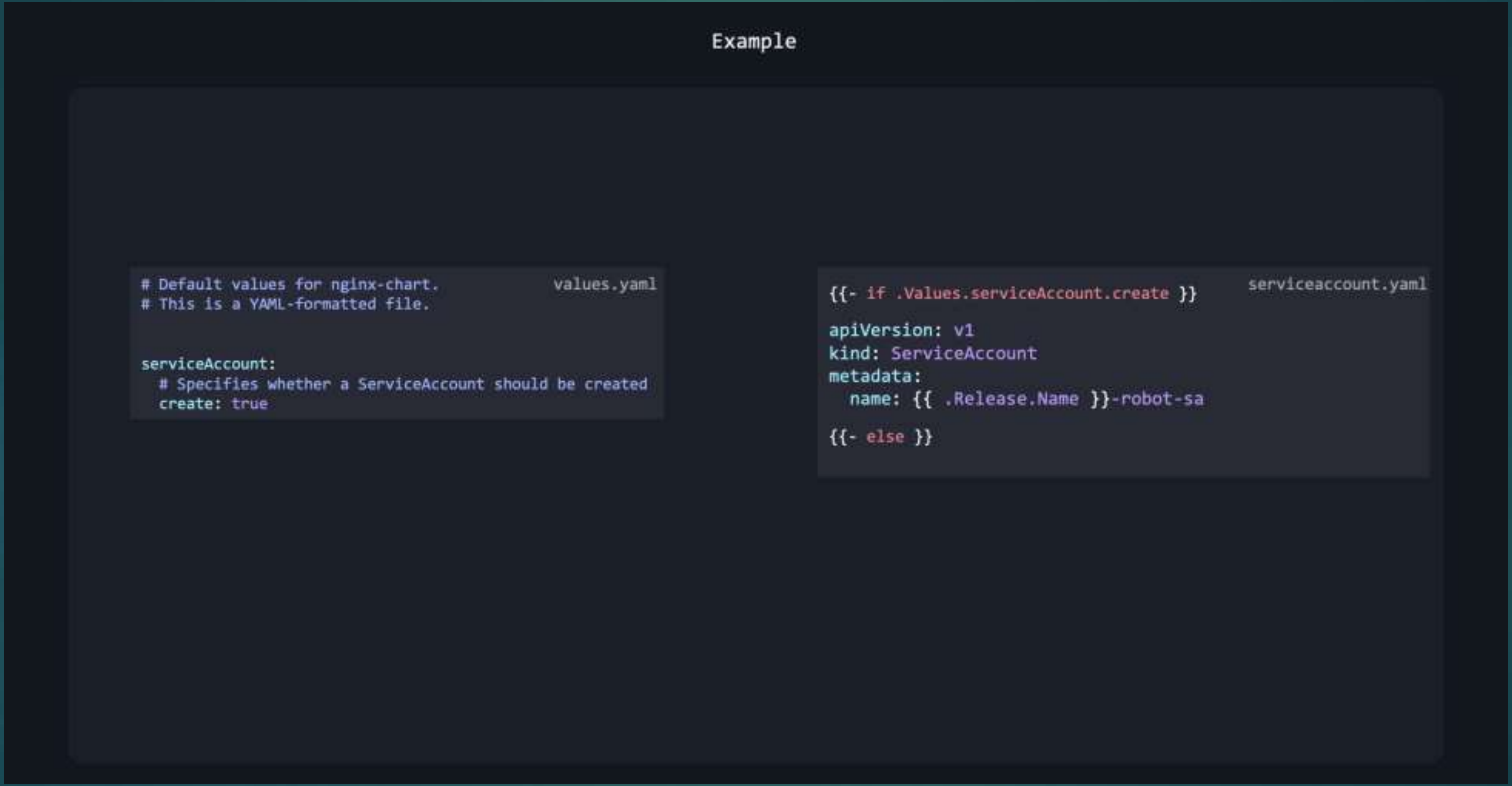


Conditionals

service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-nginx
  {{- if .Values.orgLabel }}
  labels:
    org: {{ .Values.orgLabel }}
  {{- else if eq .Values.orgLabel "hr" }}
  labels:
    org: human resources
  {{- end }}
spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```

Function	Purpose
eq	equal
ne	not equal
lt	less than
le	less than or equal to
gt	greater than
ge	greater than or equal to
not	negation
empty	value is empty



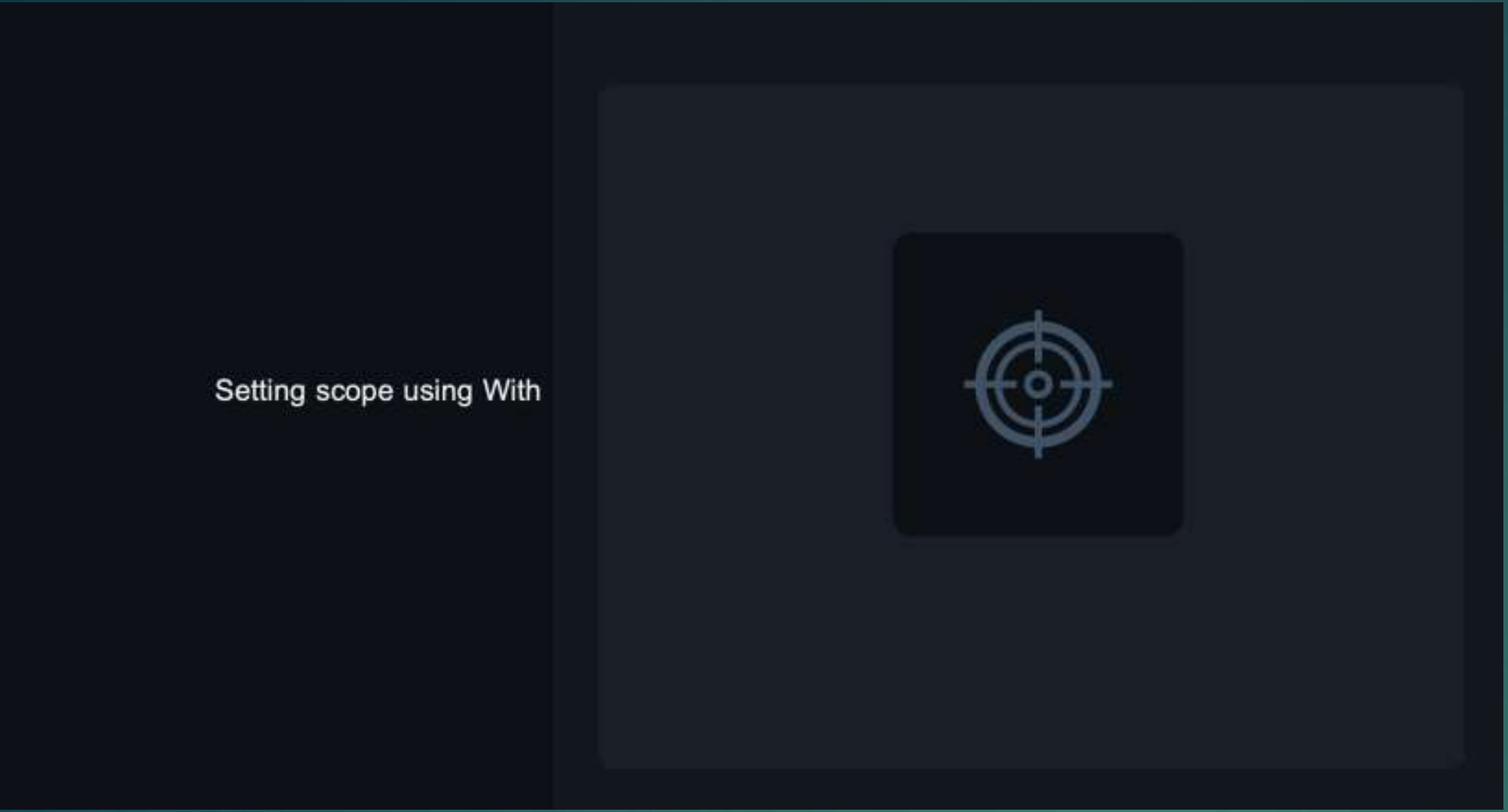
Example

values.yaml

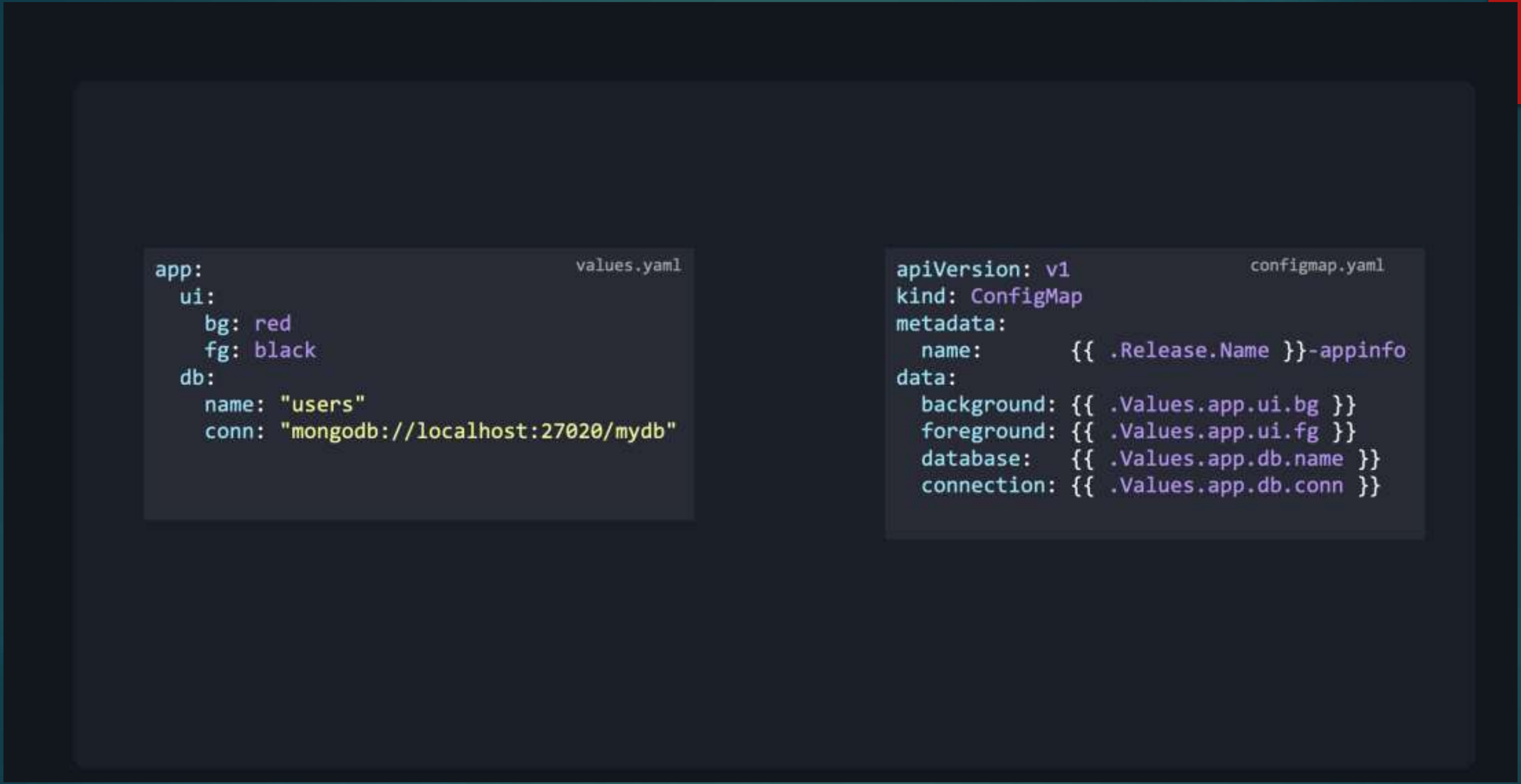
```
# Default values for nginx-chart.
# This is a YAML-formatted file.


serviceAccount:
  # Specifies whether a ServiceAccount should be created
  create: true
```

serviceaccount.yaml

```
{{- if .Values.serviceAccount.create }}

apiVersion: v1
kind: ServiceAccount
metadata:
  name: {{ .Release.Name }}-robot-sa

{{- else }}
```



Setting scope using With

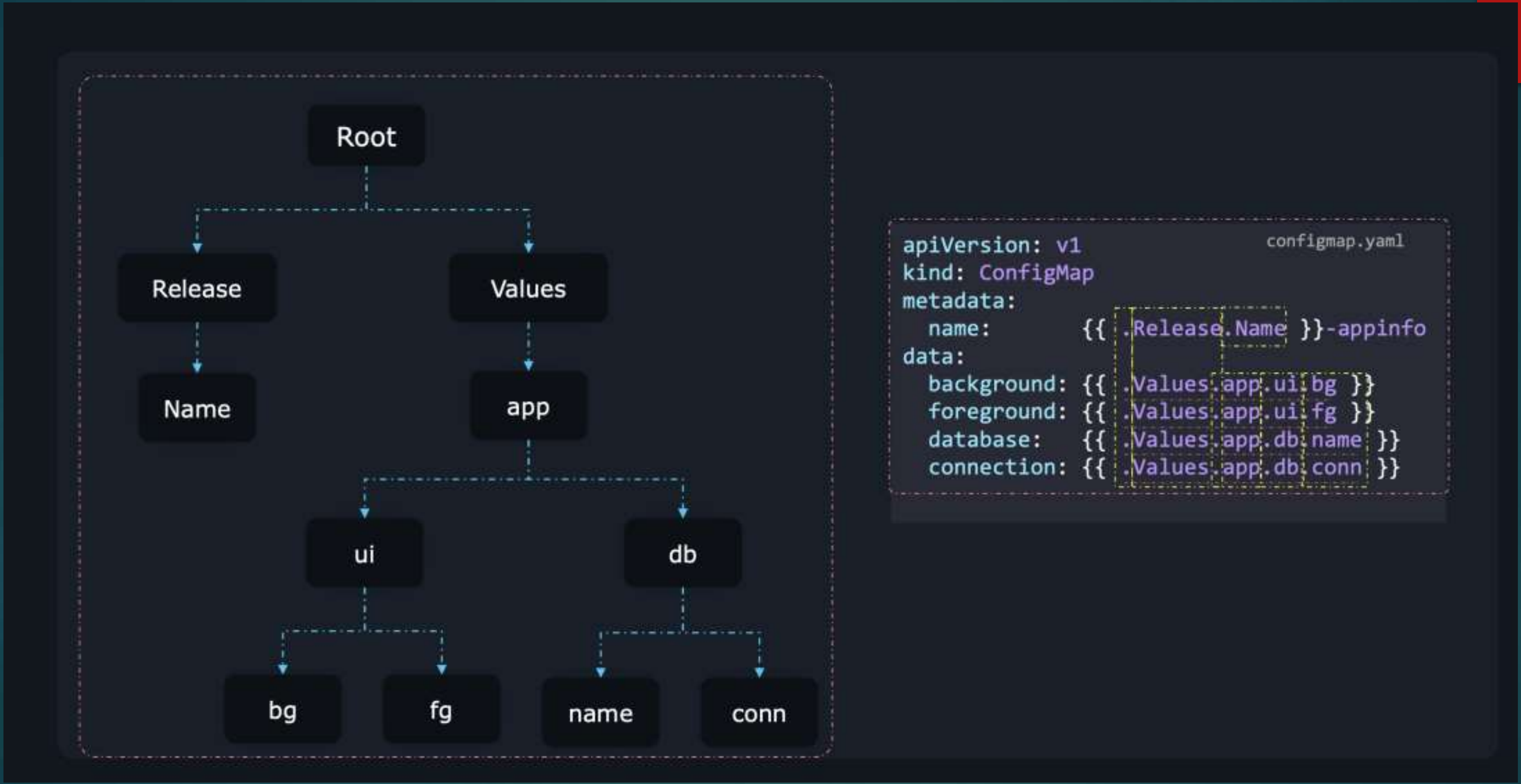


values.yaml

```yaml
app:
  ui:
    bg: red
    fg: black
  db:
    name: "users"
    conn: "mongodb://localhost:27020/mydb"
```

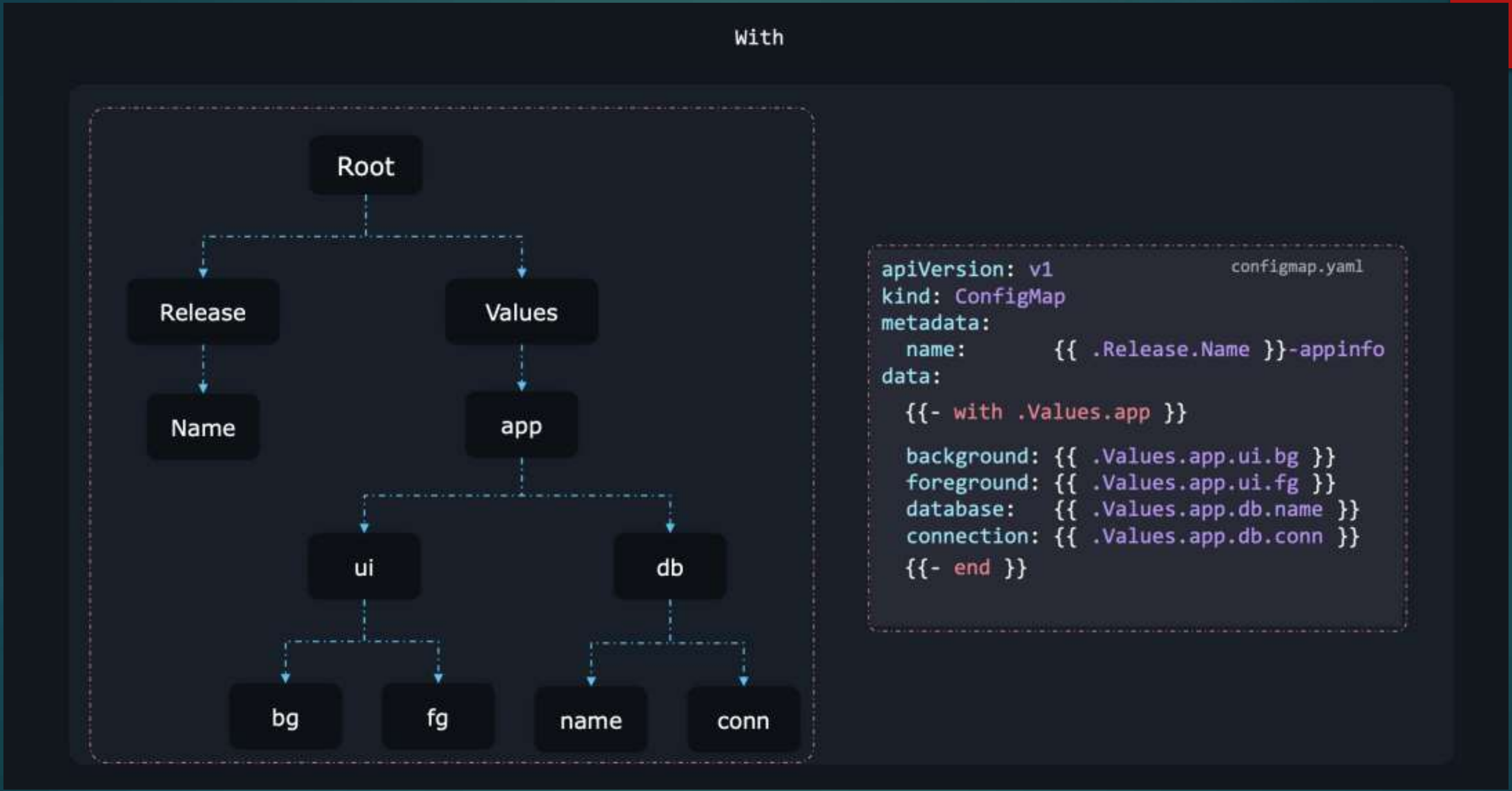
configmap.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  background: {{ .Values.app.ui.bg }}
  foreground: {{ .Values.app.ui.fg }}
  database:   {{ .Values.app.db.name }}
  connection: {{ .Values.app.db.conn }}
```



configmap.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  background: {{ .Values.app.ui.bg }}
  foreground: {{ .Values.app.ui.fg }}
  database:   {{ .Values.app.db.name }}
  connection: {{ .Values.app.db.conn }}
```



With

Root

Release

Values

Name

app

ui

db

bg

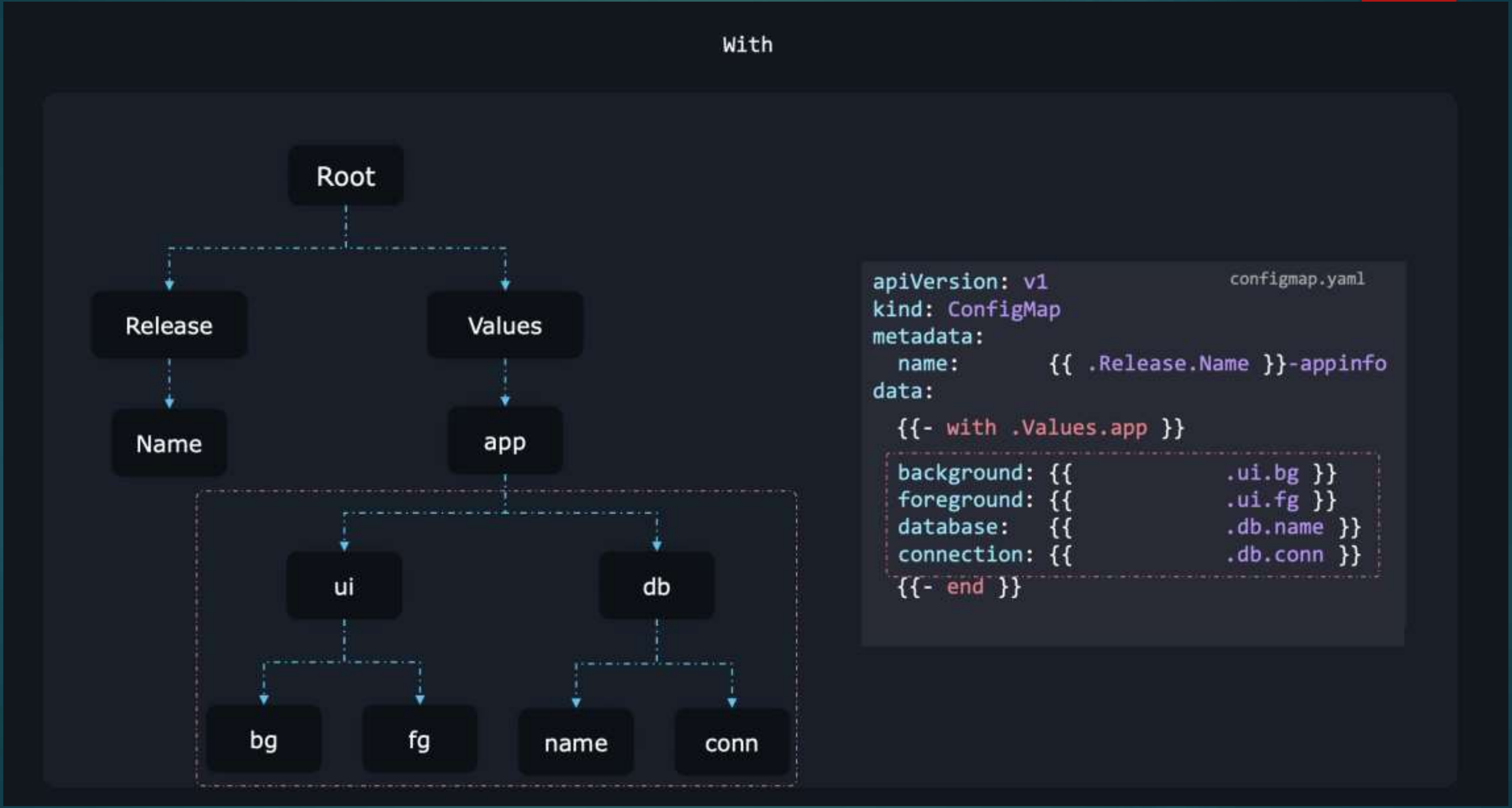
fg

name

conn

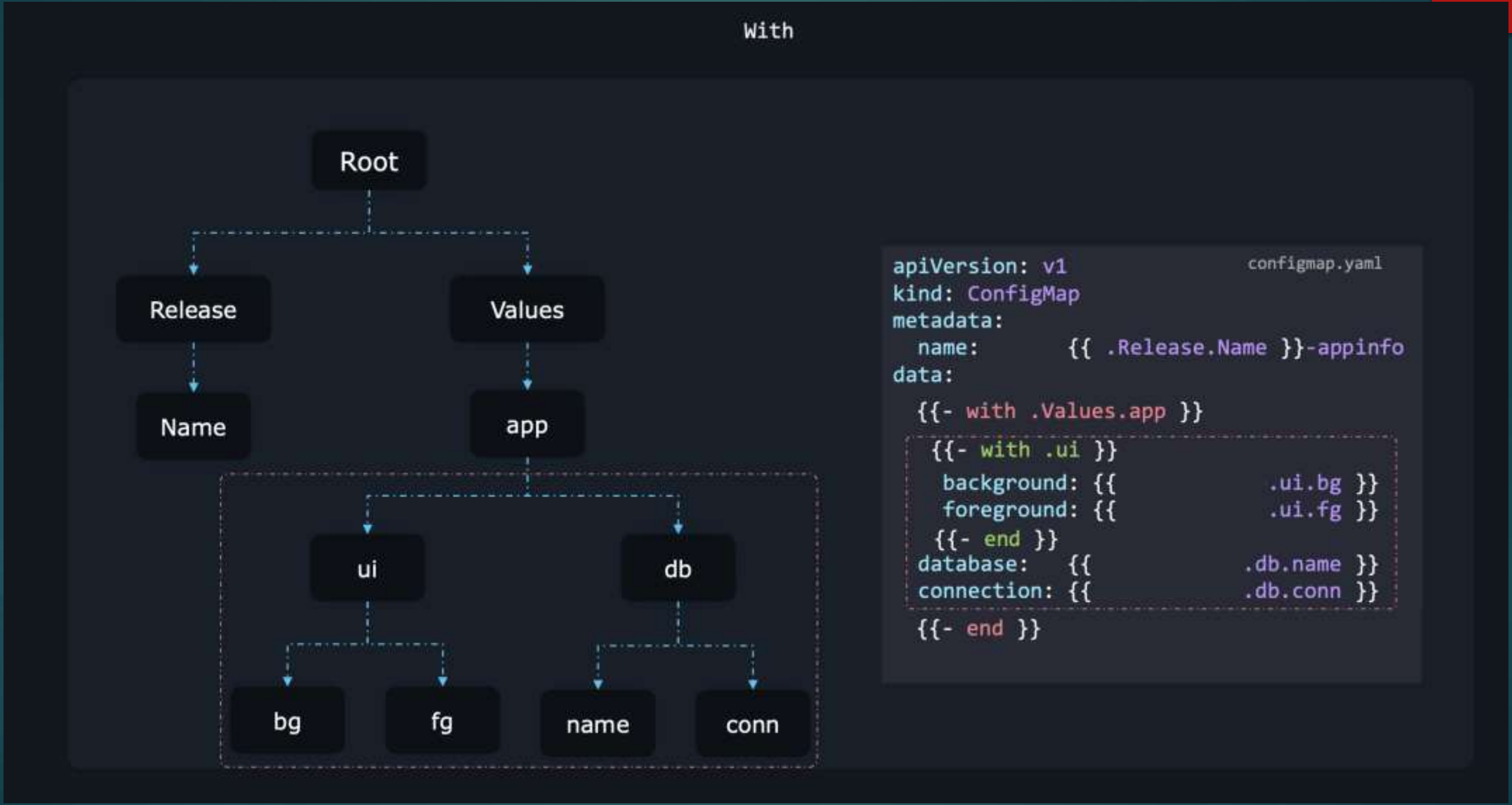
configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  {{- with .Values.app }}
  background: {{ .Values.app.ui.bg }}
  foreground: {{ .Values.app.ui.fg }}
  database:   {{ .Values.app.db.name }}
  connection: {{ .Values.app.db.conn }}
  {{- end }}
```



configmap.yaml

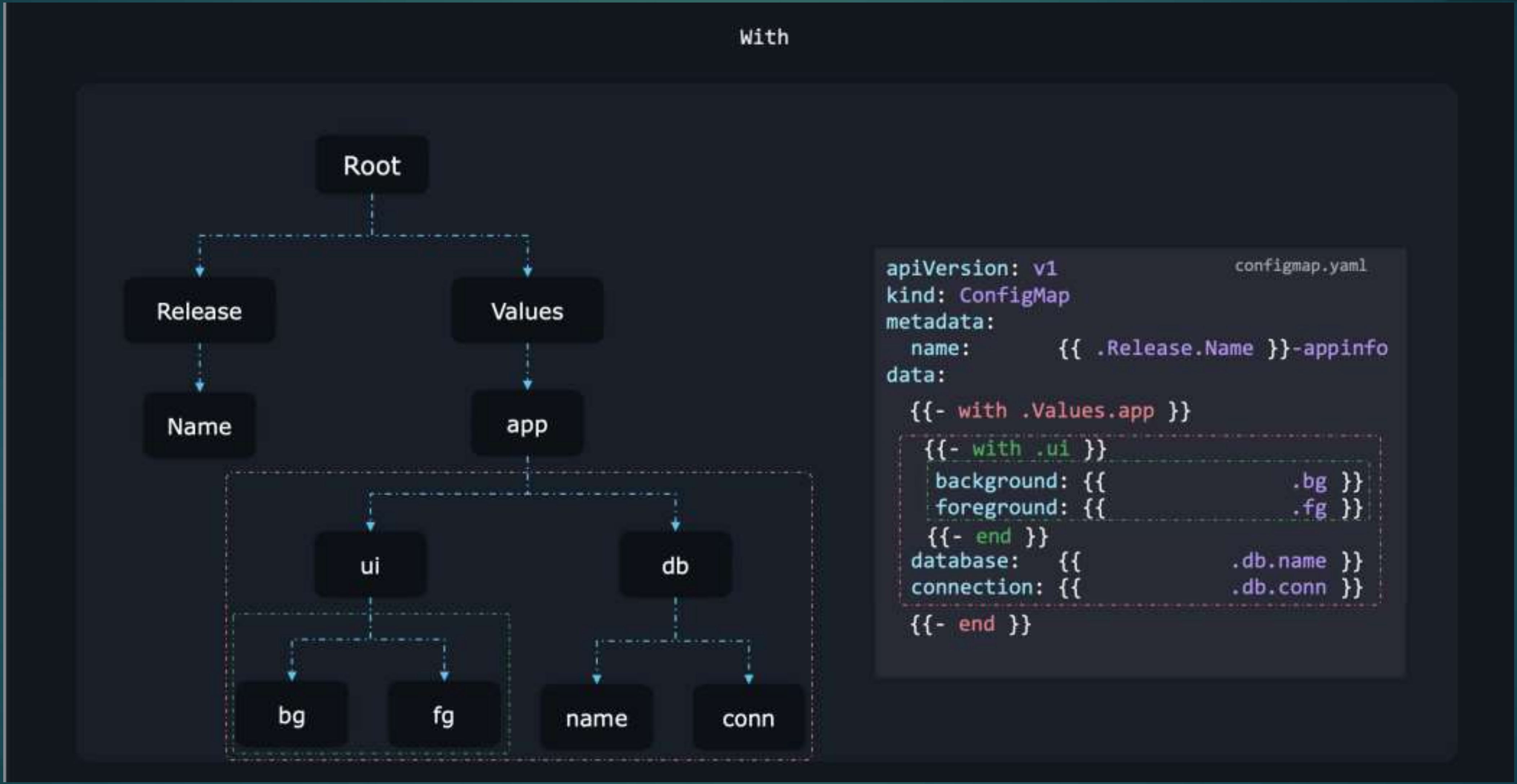
```
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  {{- with .Values.app }}
  background: {{          .ui.bg }}
  foreground: {{          .ui.fg }}
  database:   {{          .db.name }}
  connection: {{          .db.conn }}
  {{- end }}
```



With

configmap.yaml

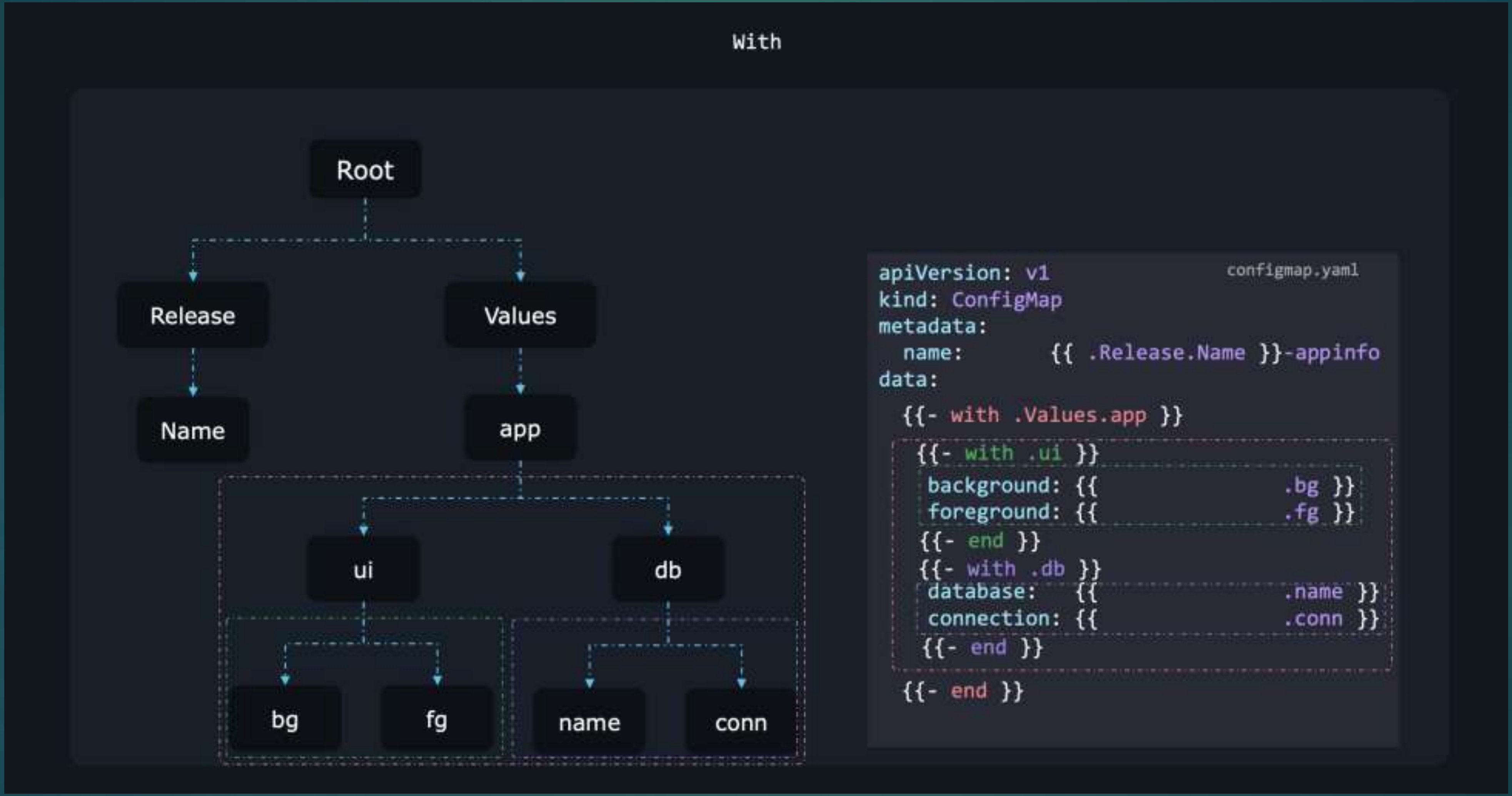
```
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  {{- with .Values.app }}
   {{- with .ui }}
    background: {{           .ui.bg }}
    foreground: {{           .ui.fg }}
   {{- end }}
  database:   {{          .db.name }}
  connection: {{          .db.conn }}
  {{- end }}
```



With

configmap.yaml

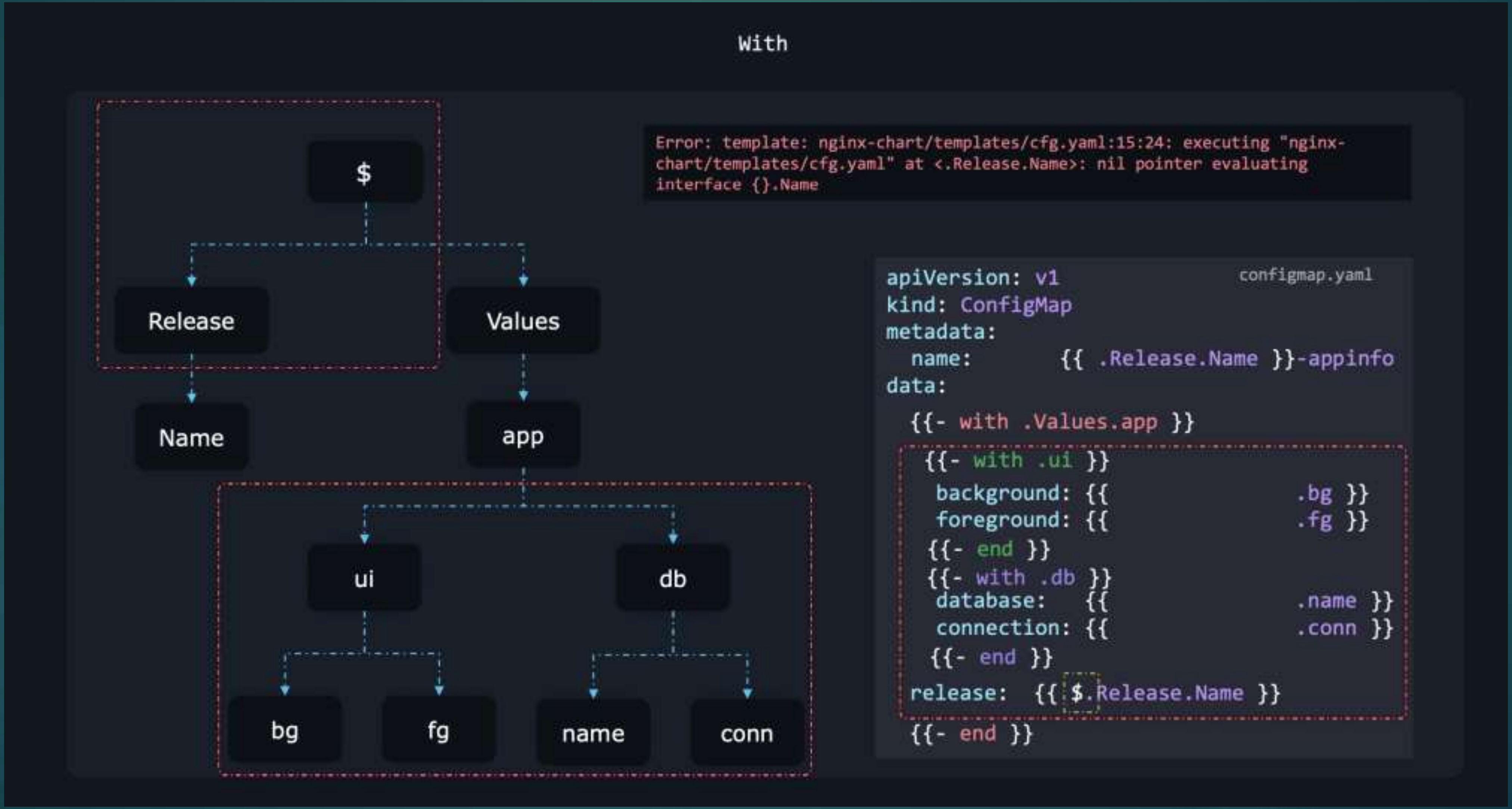
```
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  {{- with .Values.app }}
   {{- with .ui }}
    background: {{             .bg }}
    foreground: {{             .fg }}
   {{- end }}
  database:   {{          .db.name }}
  connection: {{          .db.conn }}
  {{- end }}
```



With

configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  {{- with .Values.app }}
   {{- with .ui }}
    background: {{              .bg }}
    foreground: {{              .fg }}
   {{- end }}
   {{- with .db }}
    database:   {{              .name }}
    connection: {{              .conn }}
   {{- end }}
  {{- end }}
```

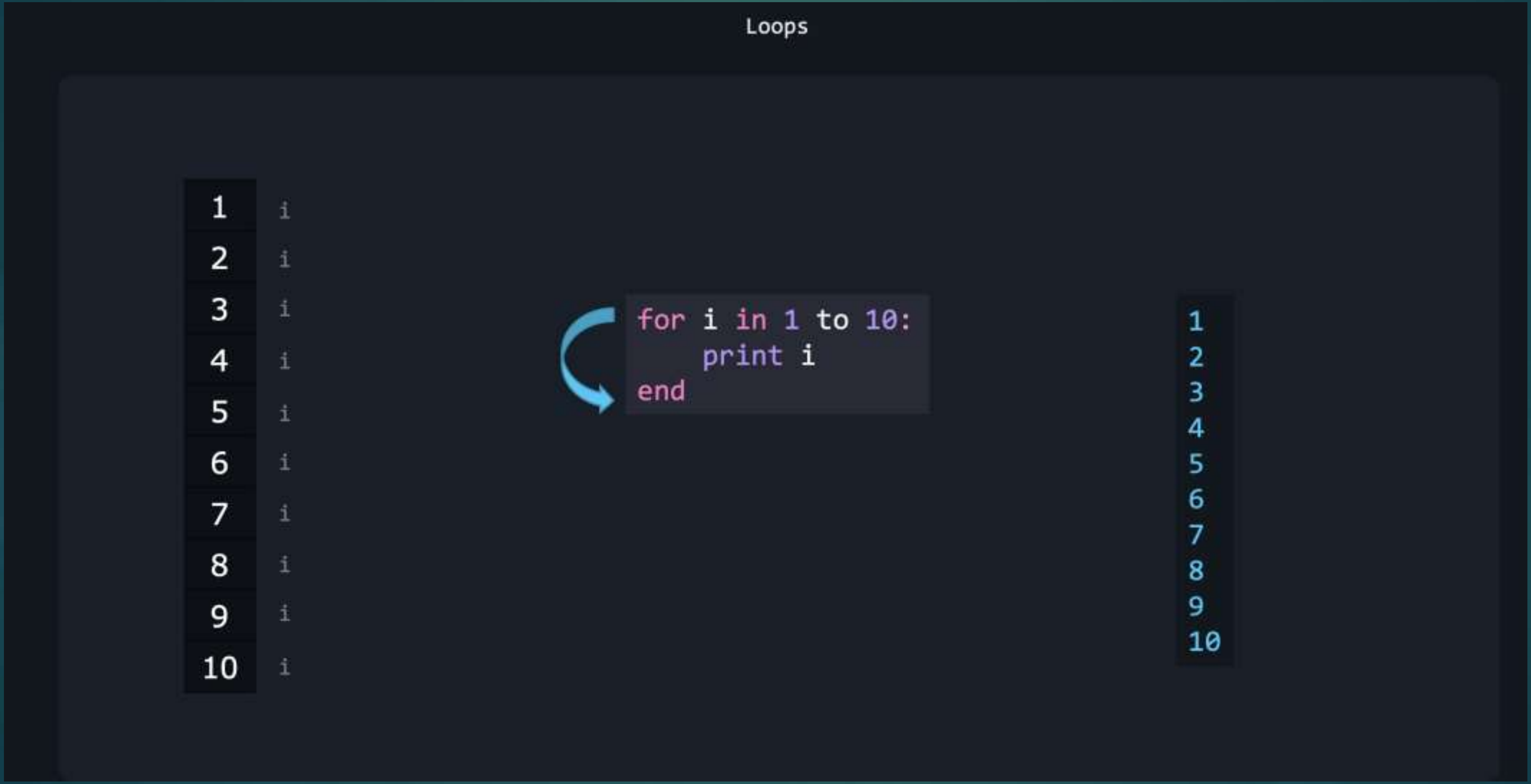


```
Error: template: nginx-chart/templates/cfg.yaml:15:24: executing "nginx-chart/templates/cfg.yaml" at <.Release.Name>: nil pointer evaluating interface {}.Name
```

configmap.yaml

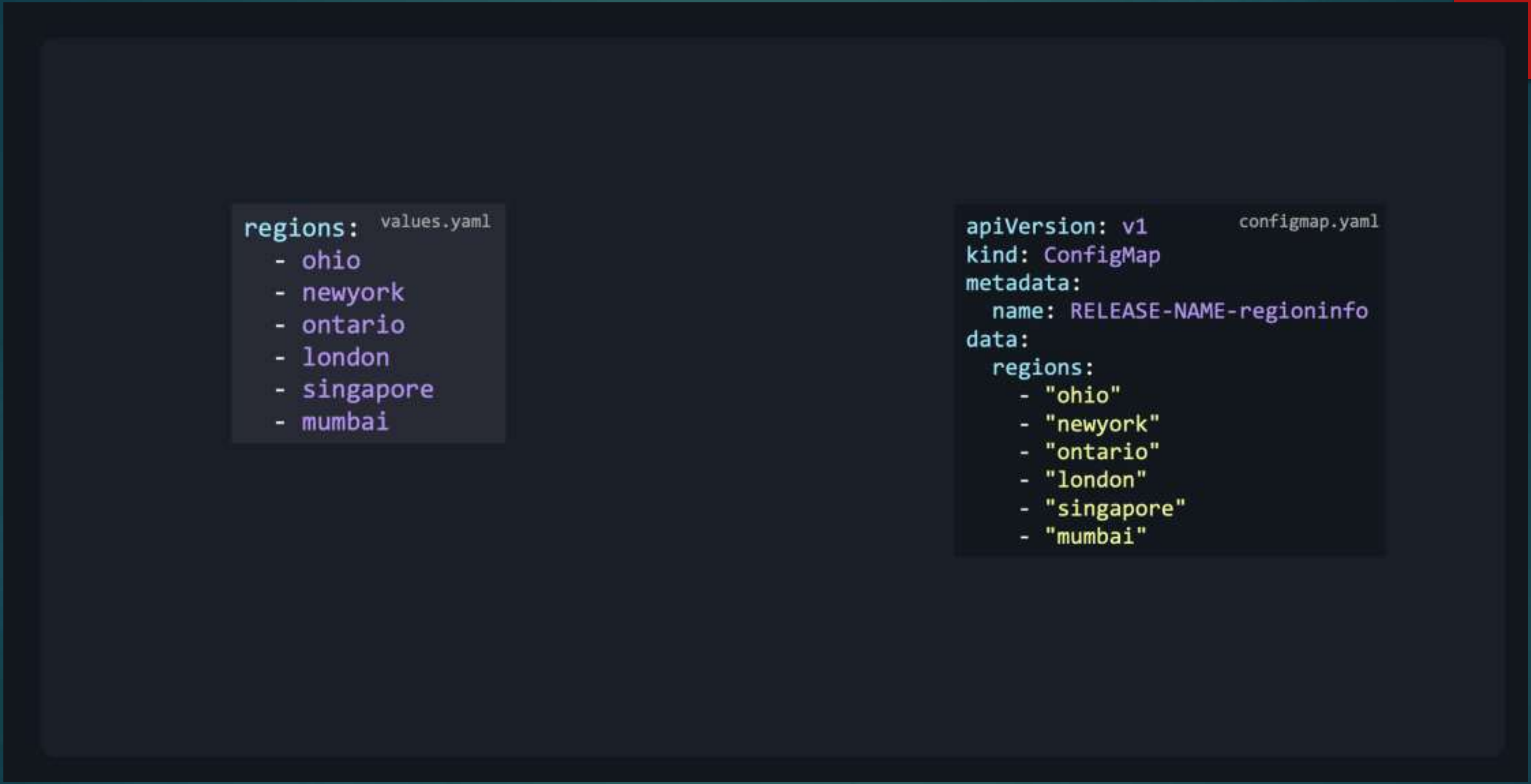
```
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  {{- with .Values.app }}
   {{- with .ui }}
    background: {{              .bg }}
    foreground: {{              .fg }}
   {{- end }}
   {{- with .db }}
    database:   {{              .name }}
    connection: {{              .conn }}
   {{- end }}
  release:  {{ $.Release.Name }}
  {{- end }}
```

Range



```
for i in 1 to 10:
    print i
end
```

```
1
2
3
4
5
6
7
8
9
10
```

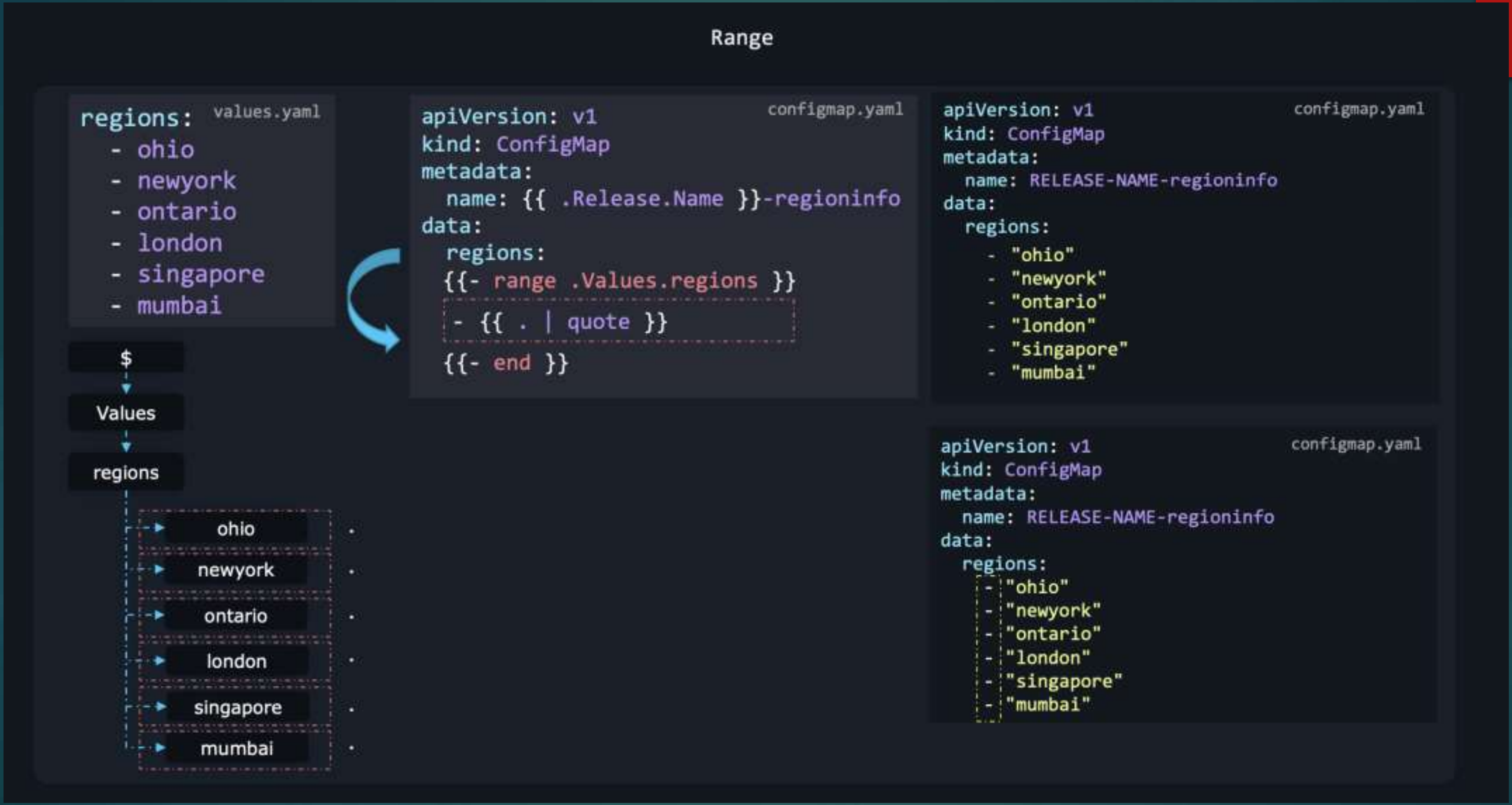


values.yaml

```yaml
regions:
  - ohio
  - newyork
  - ontario
  - london
  - singapore
  - mumbai
```

configmap.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: RELEASE-NAME-regioninfo
data:
  regions:
    - "ohio"
    - "newyork"
    - "ontario"
    - "london"
    - "singapore"
    - "mumbai"
```



Range

values.yaml

```yaml
regions:
  - ohio
  - newyork
  - ontario
  - london
  - singapore
  - mumbai
```

configmap.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-regioninfo
data:
  regions:
  {{- range .Values.regions }}
  - {{ . | quote }}
  {{- end }}
```

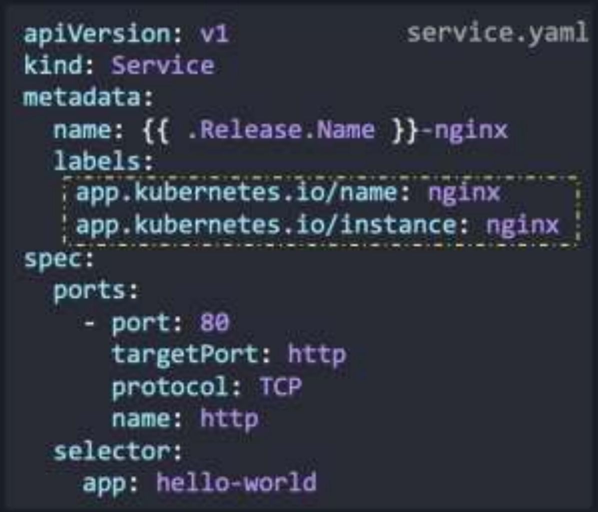
configmap.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: RELEASE-NAME-regioninfo
data:
  regions:
    - "ohio"
    - "newyork"
    - "ontario"
    - "london"
    - "singapore"
    - "mumbai"
```

configmap.yaml

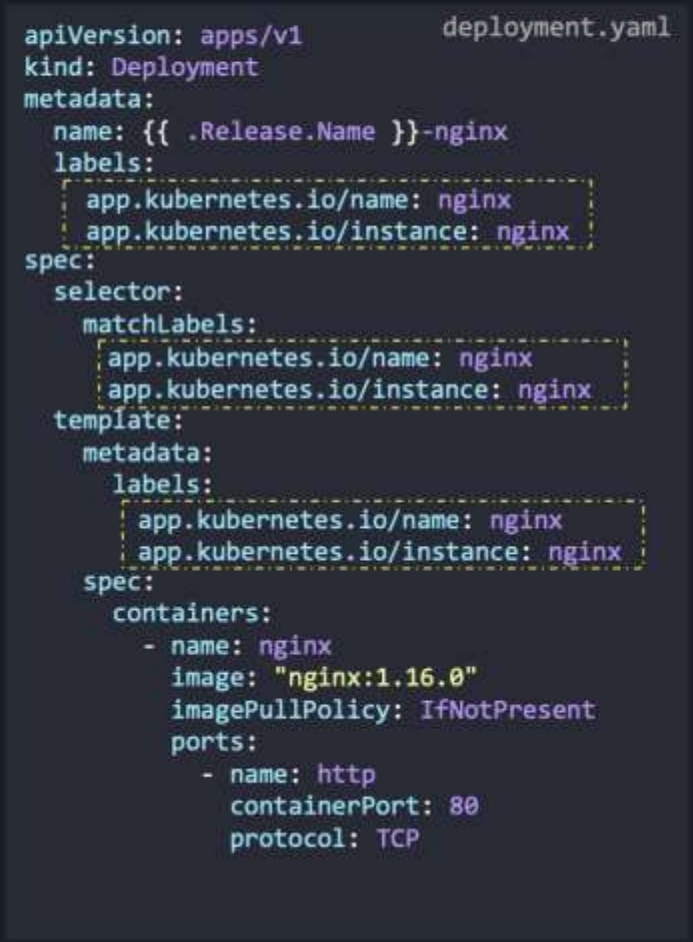
```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: RELEASE-NAME-regioninfo
data:
  regions:
    - "ohio"
    - "newyork"
    - "ontario"
    - "london"
    - "singapore"
    - "mumbai"
```

Named Templates



service.yaml

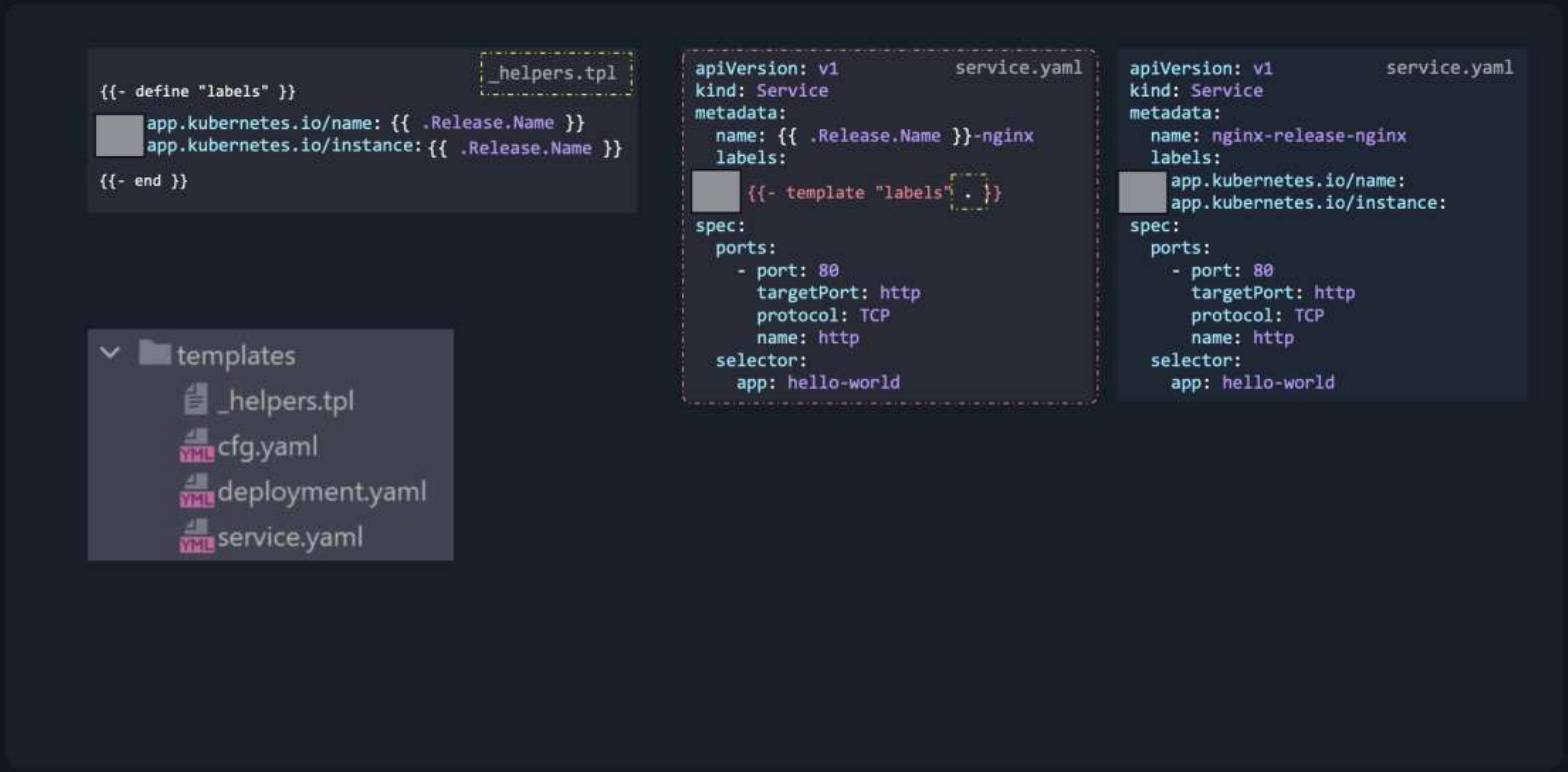
```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-nginx
  labels:
   app.kubernetes.io/name: nginx
   app.kubernetes.io/instance: nginx
spec:
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```



deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
  labels:
    app.kubernetes.io/name: nginx
    app.kubernetes.io/instance: nginx
spec:
  selector:
    matchLabels:
     app.kubernetes.io/name: nginx
     app.kubernetes.io/instance: nginx
  template:
    metadata:
      labels:
       app.kubernetes.io/name: nginx
       app.kubernetes.io/instance: nginx
    spec:
      containers:
        - name: nginx
          image: "nginx:1.16.0"
          imagePullPolicy: IfNotPresent
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

Template



_helpers.tpl

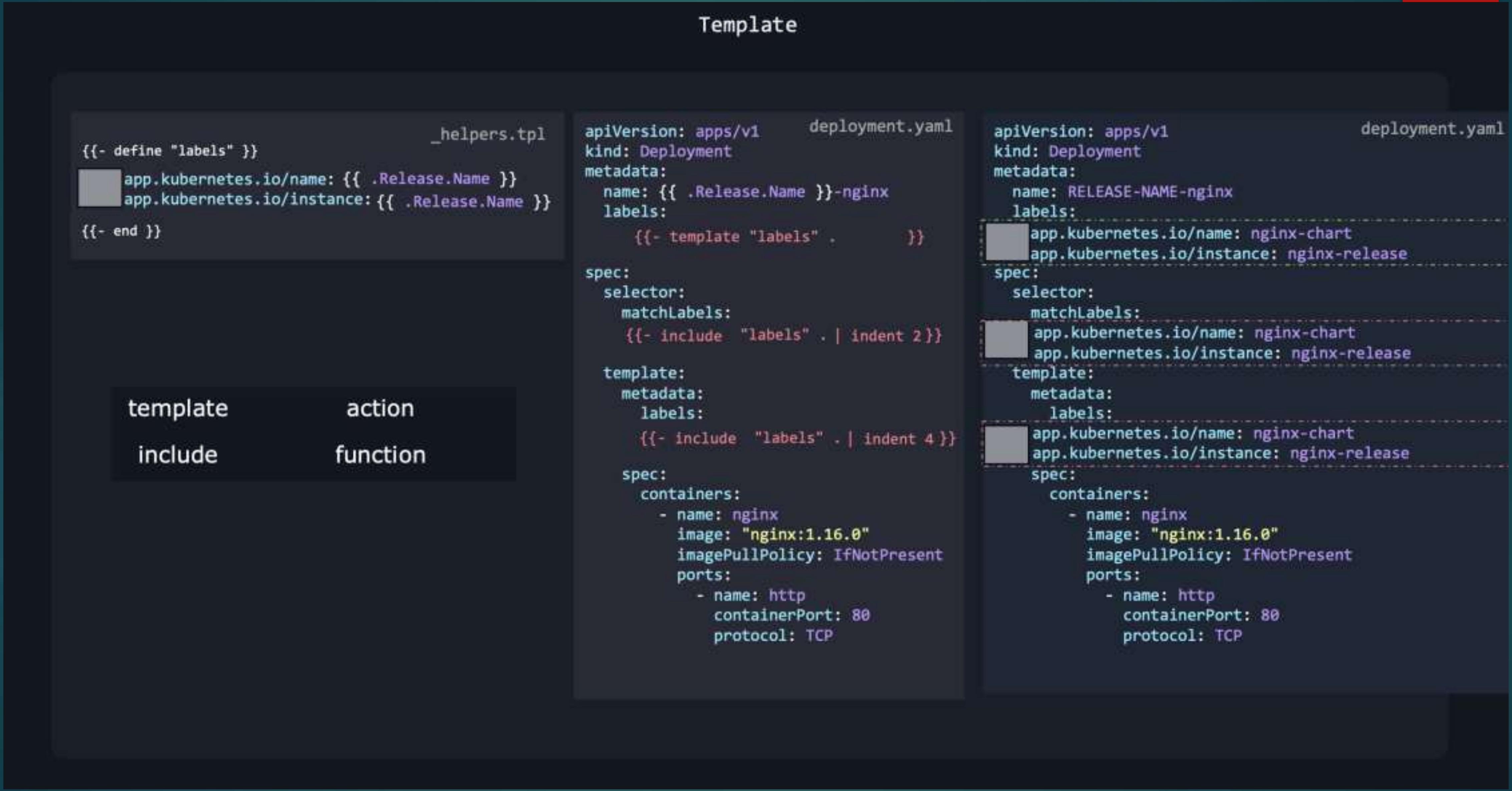
```
{{- define "labels" }}
   app.kubernetes.io/name: {{ .Release.Name }}
   app.kubernetes.io/instance: {{ .Release.Name }}
{{- end }}
```

service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-nginx
  labels:
     {{- template "labels" . }}
spec:
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-release-nginx
  labels:
     app.kubernetes.io/name:
     app.kubernetes.io/instance:
spec:
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```



Template

_helpers.tpl

```
{{- define "labels" }}
app.kubernetes.io/name: {{ .Release.Name }}
app.kubernetes.io/instance: {{ .Release.Name }}
{{- end }}
```

template	action
include	function

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
  labels:
    {{- template "labels" .        }}
spec:
  selector:
    matchLabels:
    {{- include  "labels" . | indent 2}}
  template:
    metadata:
      labels:
      {{- include  "labels" . | indent 4 }}
    spec:
      containers:
        - name: nginx
          image: "nginx:1.16.0"
          imagePullPolicy: IfNotPresent
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: RELEASE-NAME-nginx
  labels:
    app.kubernetes.io/name: nginx-chart
    app.kubernetes.io/instance: nginx-release
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: nginx-chart
      app.kubernetes.io/instance: nginx-release
  template:
    metadata:
      labels:
        app.kubernetes.io/name: nginx-chart
        app.kubernetes.io/instance: nginx-release
    spec:
      containers:
        - name: nginx
          image: "nginx:1.16.0"
          imagePullPolicy: IfNotPresent
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

Lab1:-

1-Add bitnami helm chart repository in the controlplane node.

2-Deploy the Apache application on the cluster using the apache from the bitnami repository.
Set the release Name to: amaze-surf

3-Uninstall the apache chart release from the cluster.

4- install specfic version of nginx 1.22.0
, then update it to specfic 1.23.1
, then rollback

LAB2:-

Build web nginx app have index.html as configmap , also have service account (if provided the value in values file),create node port service and run on port 30310,
Also this should have PV hostpath, PVC , make sure you r using functions + loops +conditional , also make sure that all in webapp namespace