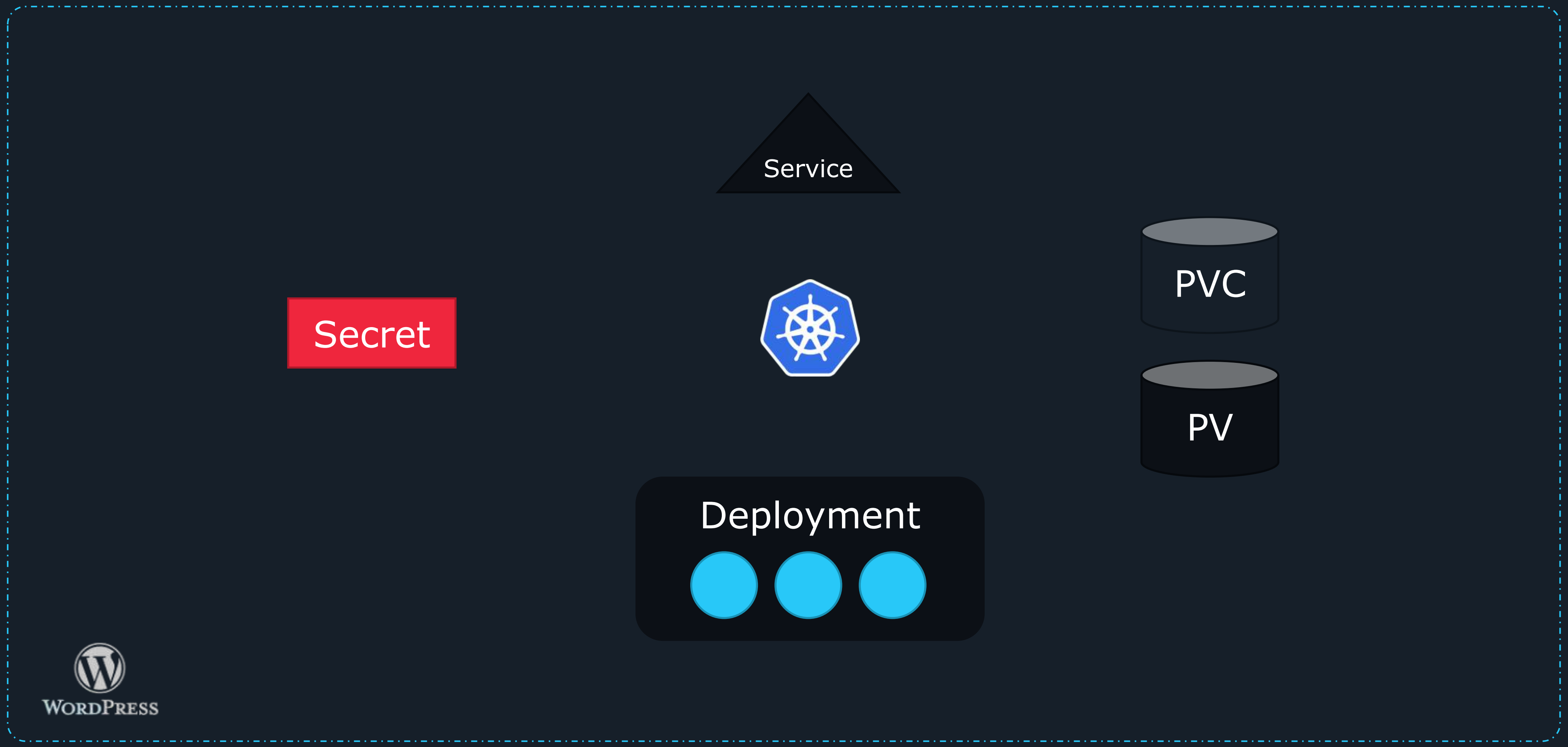
HELM

What is Helm?



Service

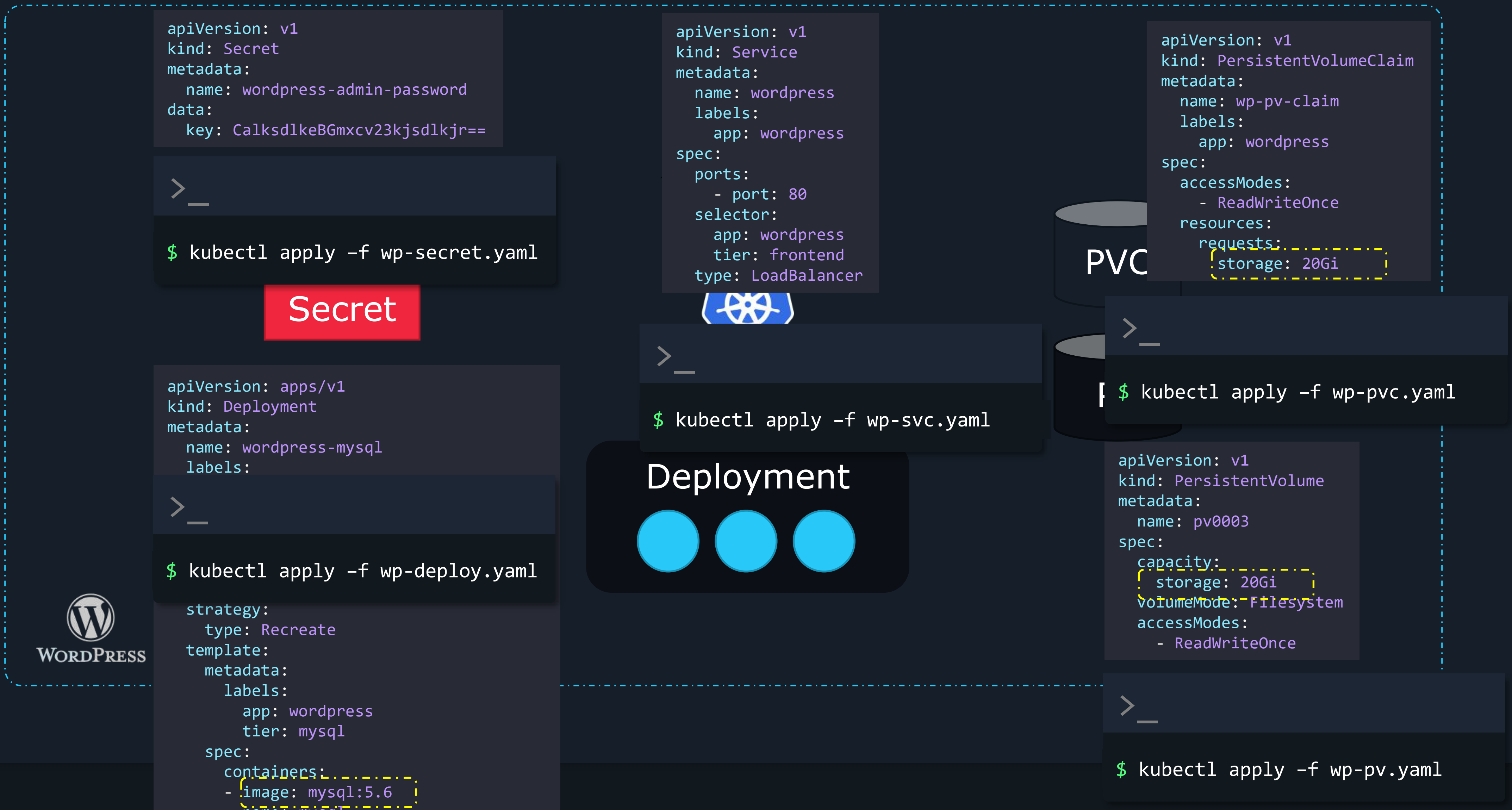
Secret

PVC

PV

Deployment

WordPress



```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
```

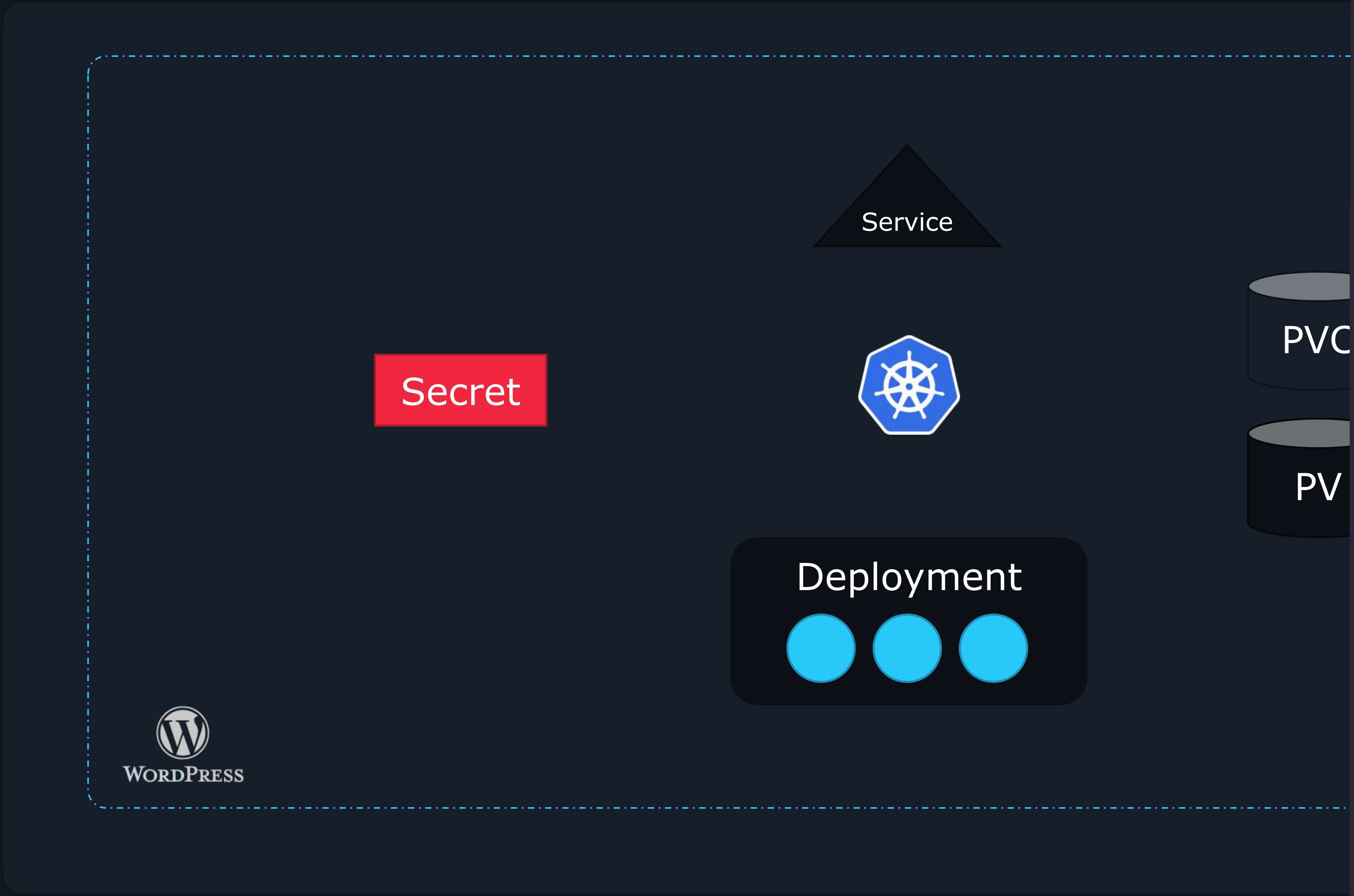
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
```

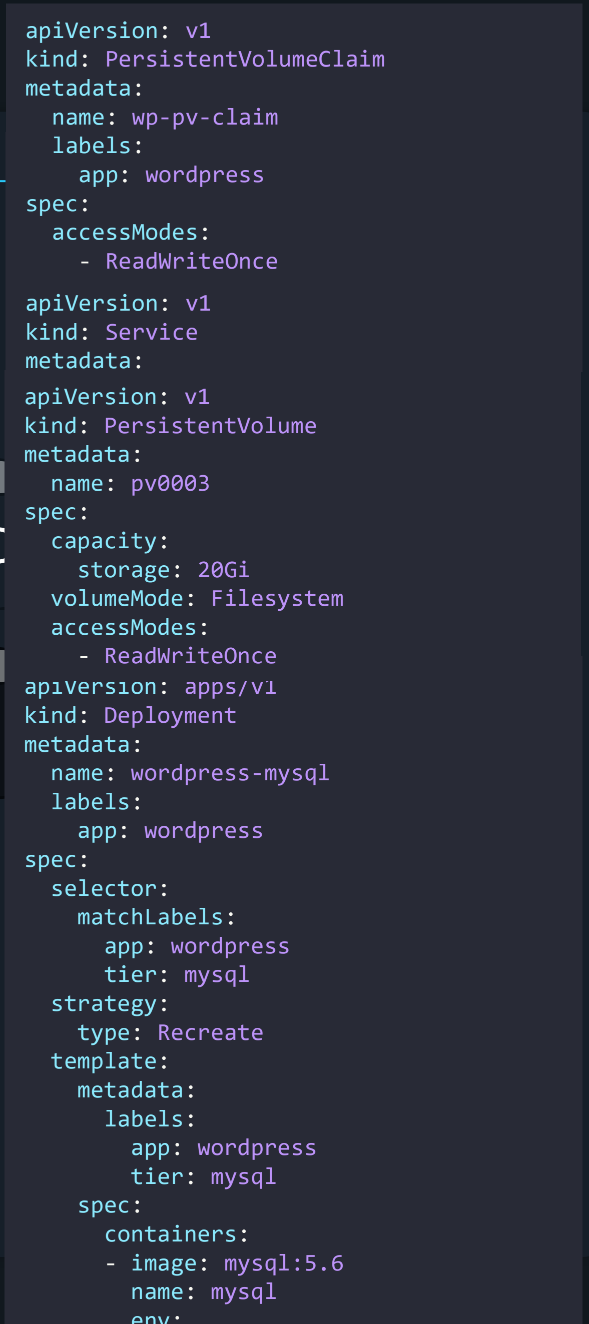
```
>_
$ kubectl apply -f wp-deploy.yaml
```

```yaml
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
    spec:
      containers:
      - image: mysql:5.6
```

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

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



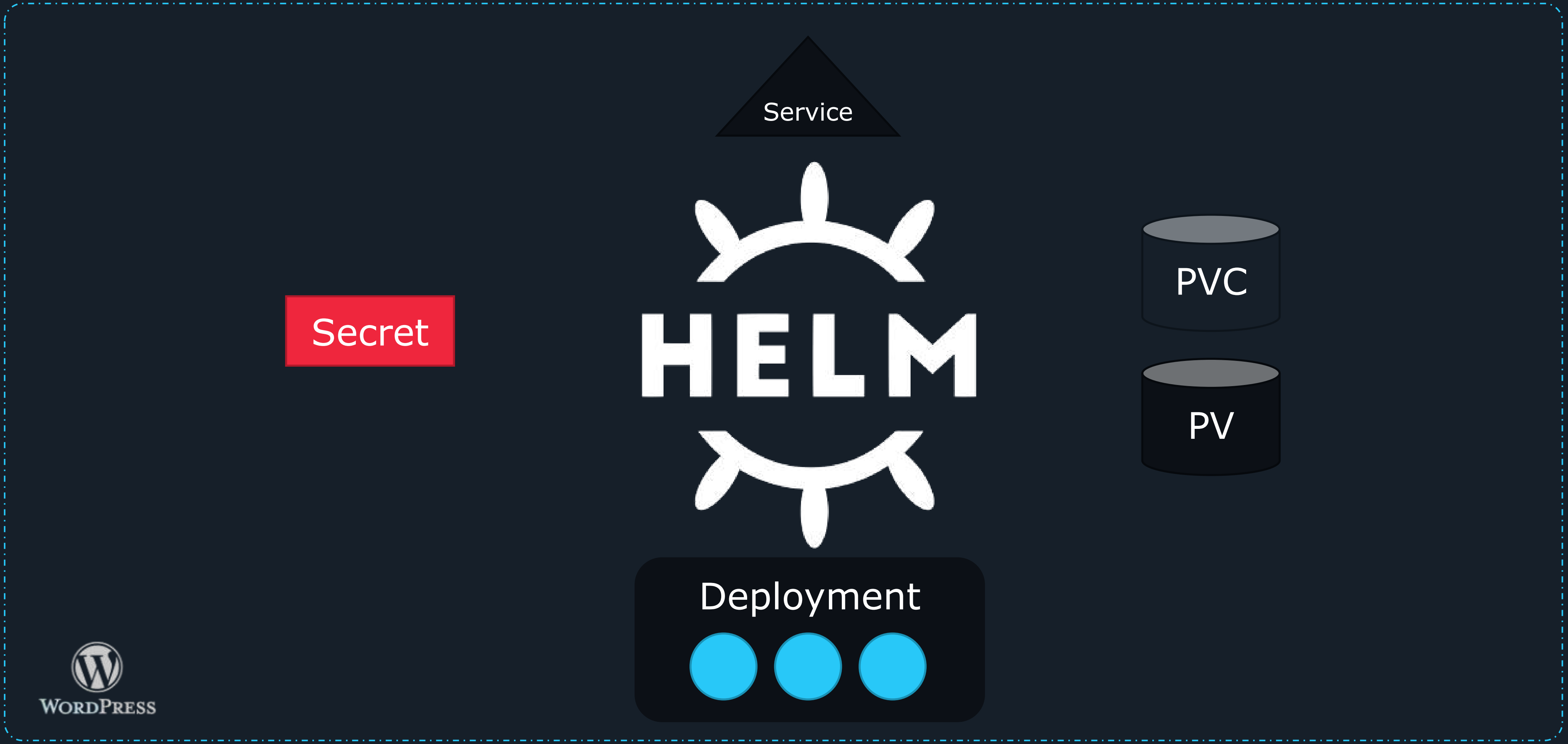


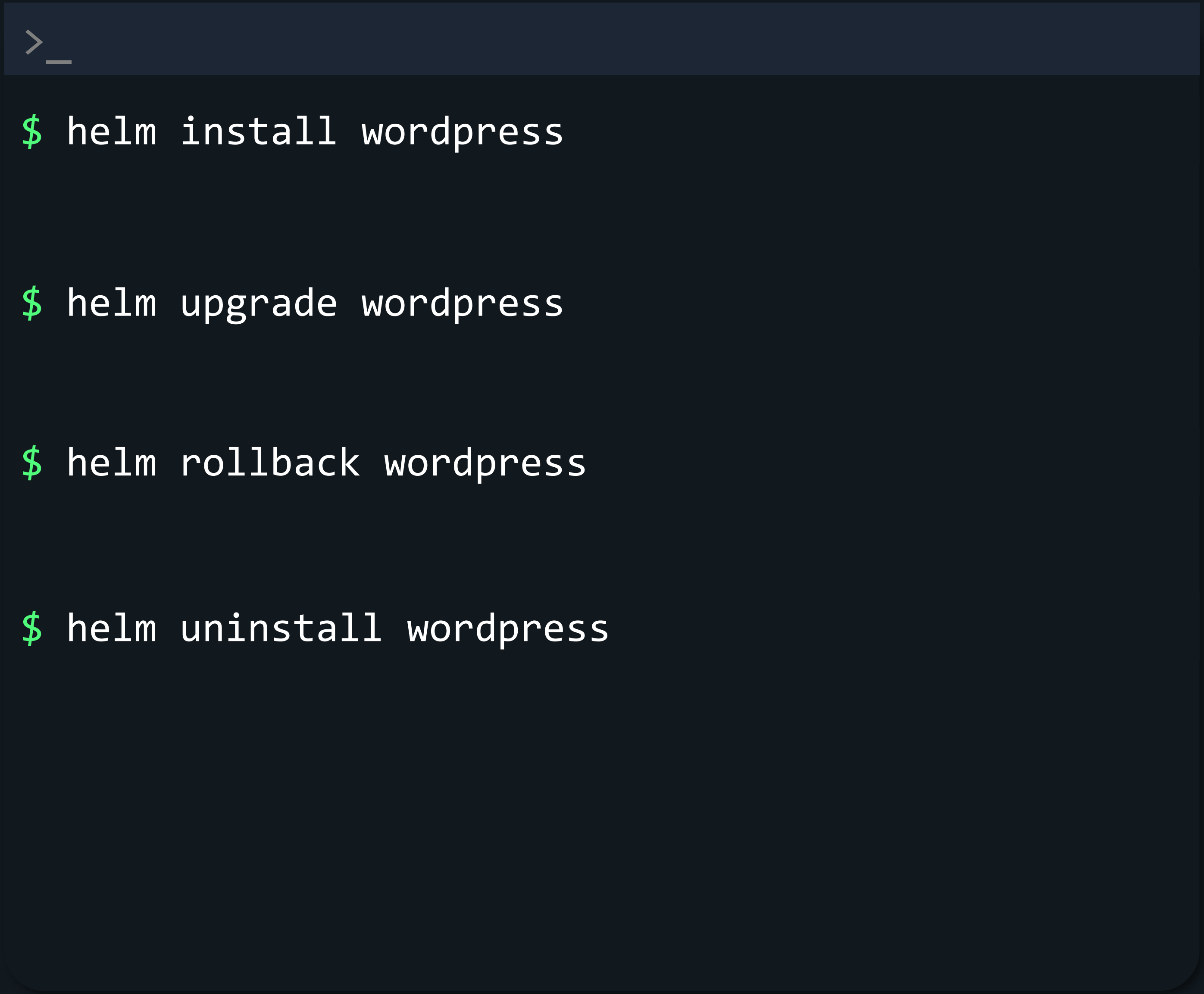
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: wordpress
spec:
  accessModes:
    - ReadWriteOnce
```

```
apiVersion: v1
kind: Service
metadata:
```

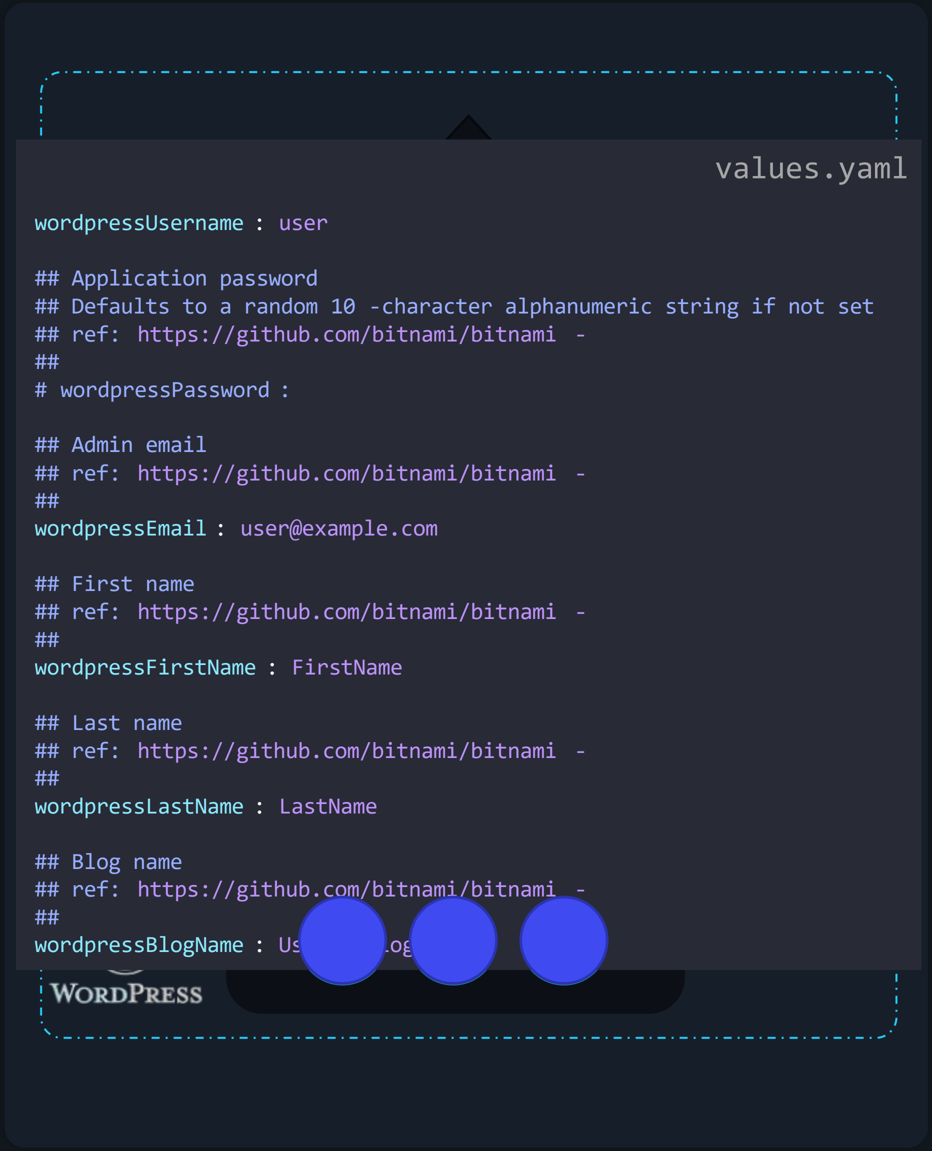
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0003
spec:
  capacity:
    storage: 20Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
```

```
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:
      containers:
      - image: mysql:5.6
        name: mysql
        env:
```


Service
Secret
HELM
PVC
PV
Deployment
WordPress



```
$ helm install wordpress

$ helm upgrade wordpress

$ helm rollback wordpress

$ helm uninstall wordpress
```

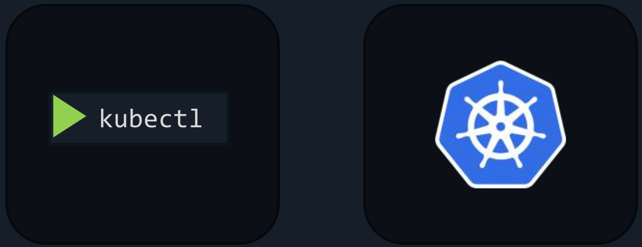


values.yaml

```
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 : Us      og
```

Installing Helm

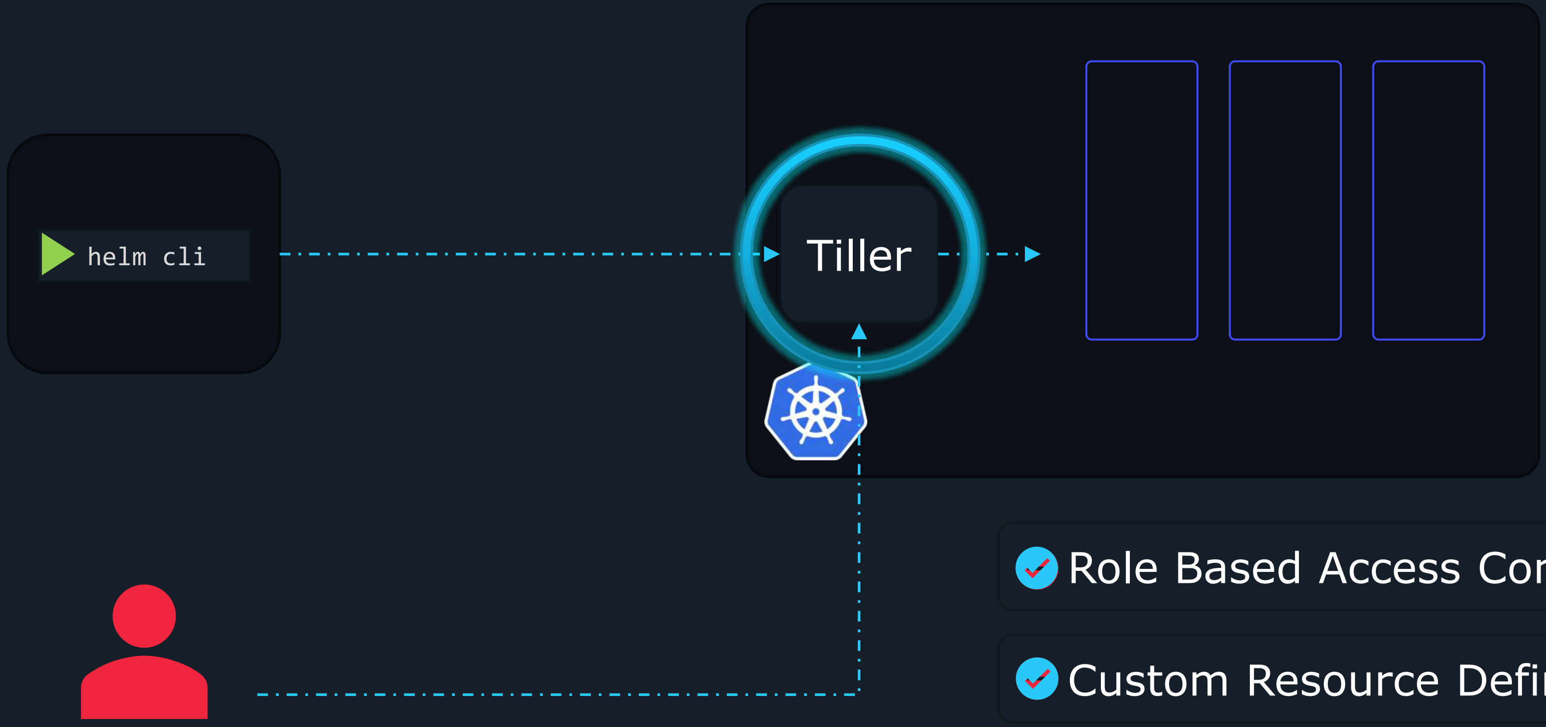
```
>_

$ 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 3

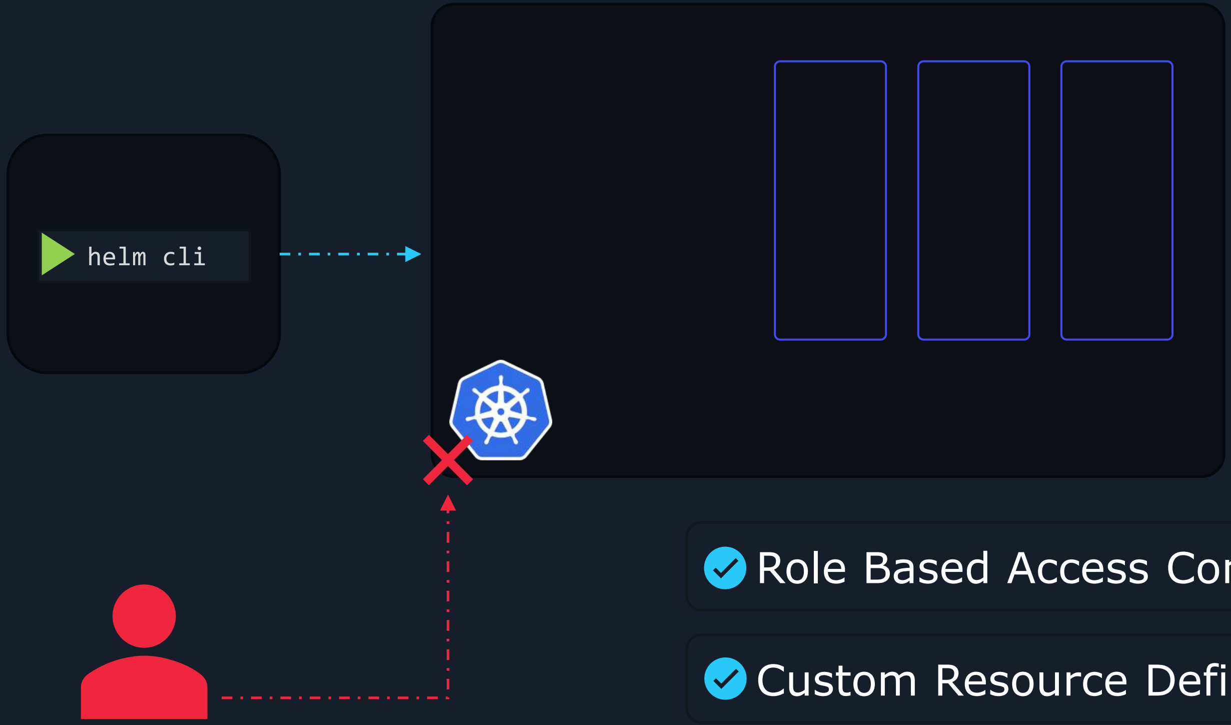
Helm 2



✓ Role Based Access Control

✓ Custom Resource Definitions

Helm 3



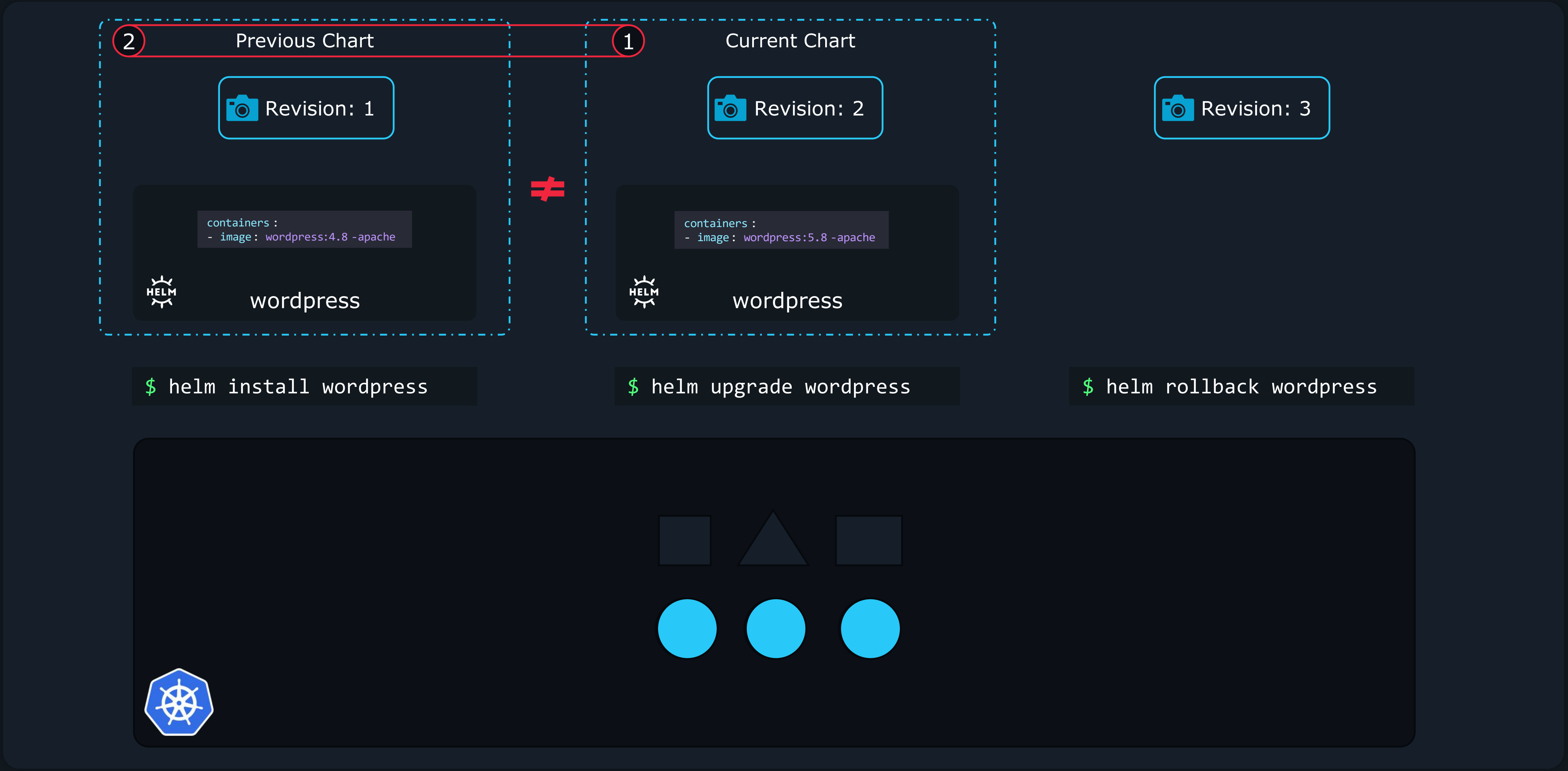
✓ Role Based Access Control

✓ Custom Resource Definitions

Helm 2 vs Helm 3

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

Helm 2



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

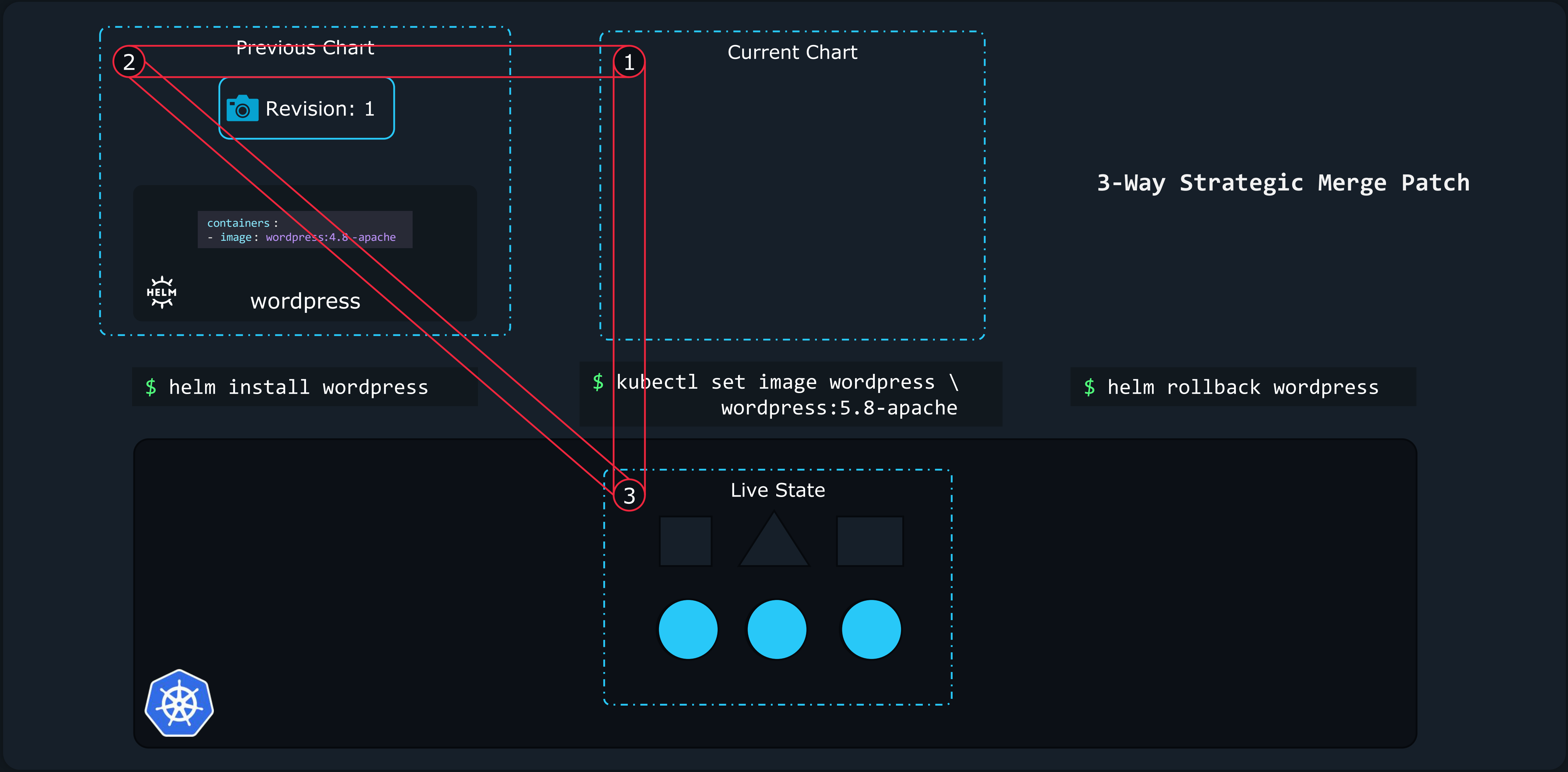
```
containers :
- image : wordpress:5.8 -apache
```

```
$ helm install wordpress
```

```
$ helm upgrade wordpress
```

```
$ helm rollback wordpress
```

Helm 3



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

3-Way Strategic Merge Patch

```
$ helm install wordpress
```

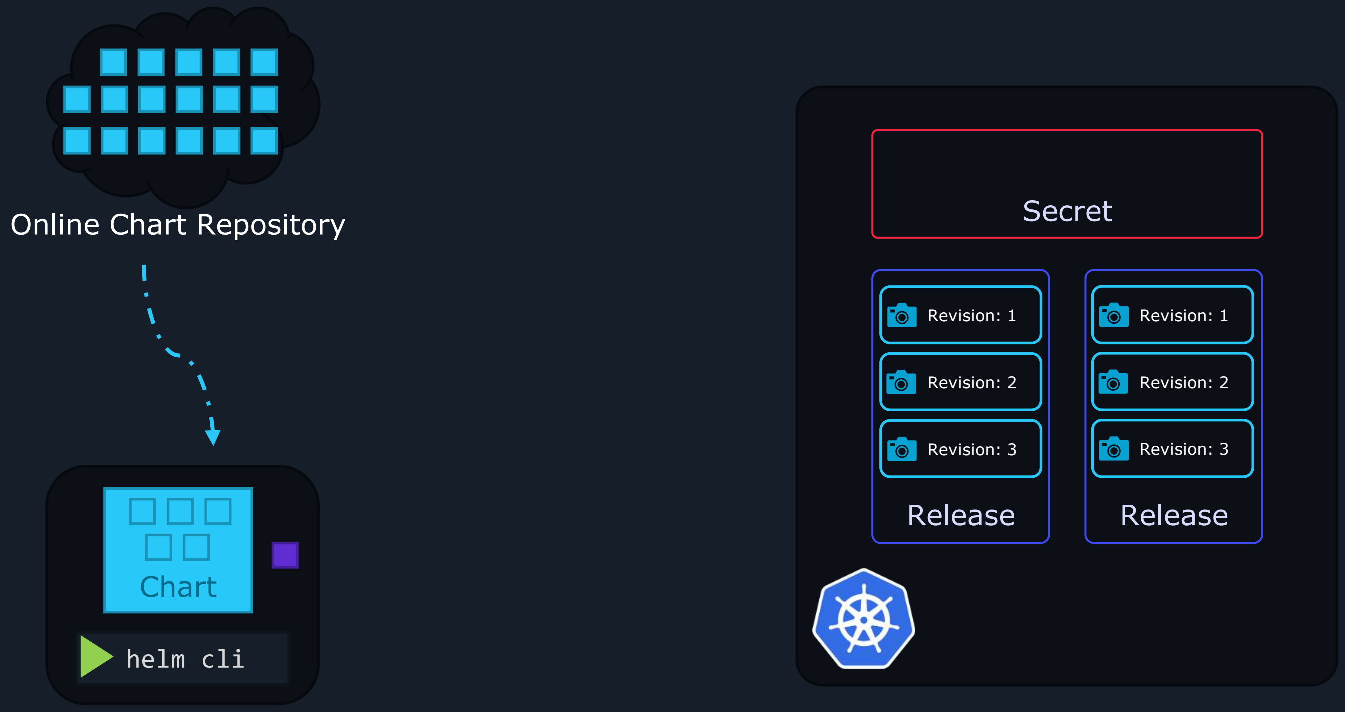
```
$ kubectl set image wordpress \
                    wordpress:5.8-apache
```

```
$ helm rollback wordpress
```

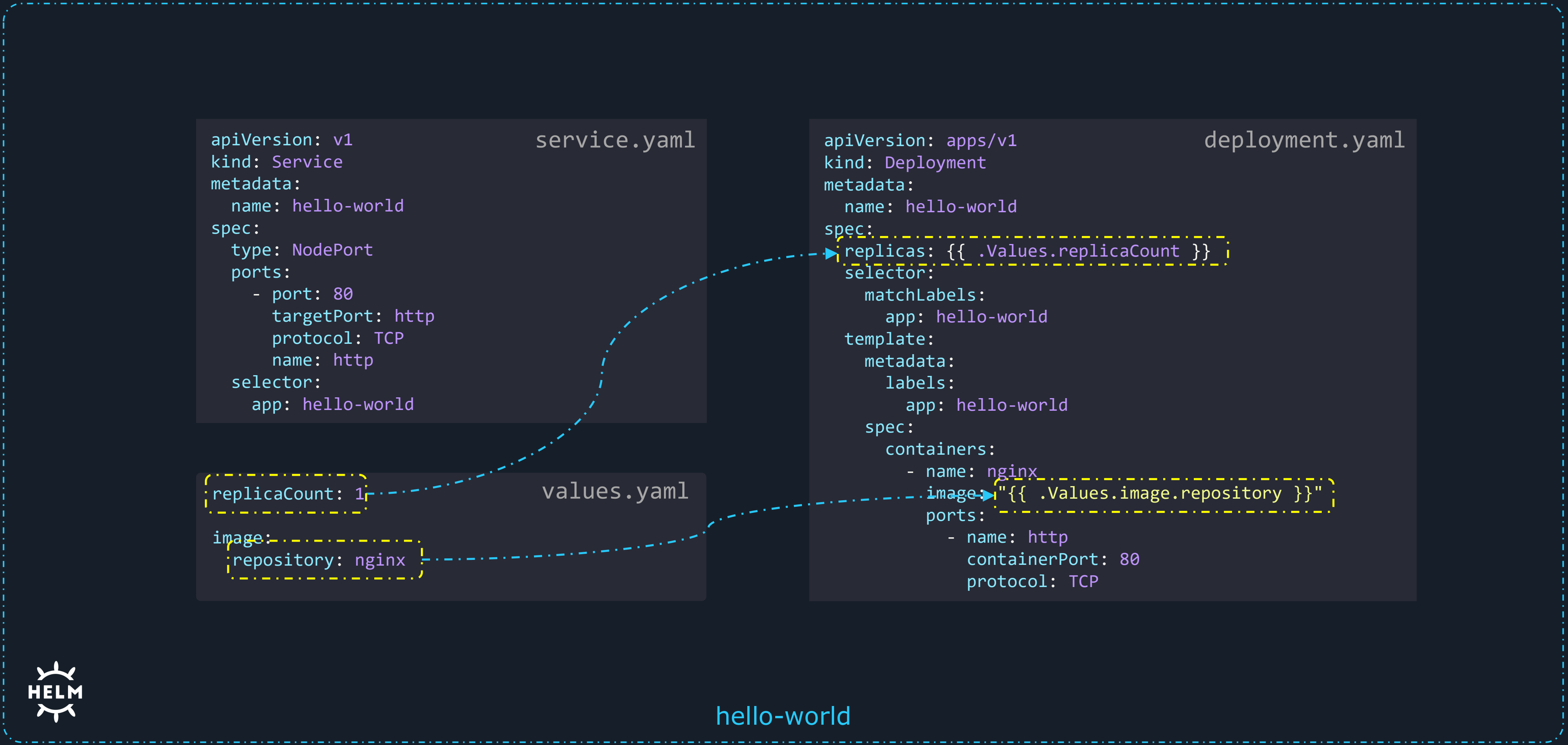
Helm Components

KODEKLOUD

Helm Components



Helm Charts



service.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
```

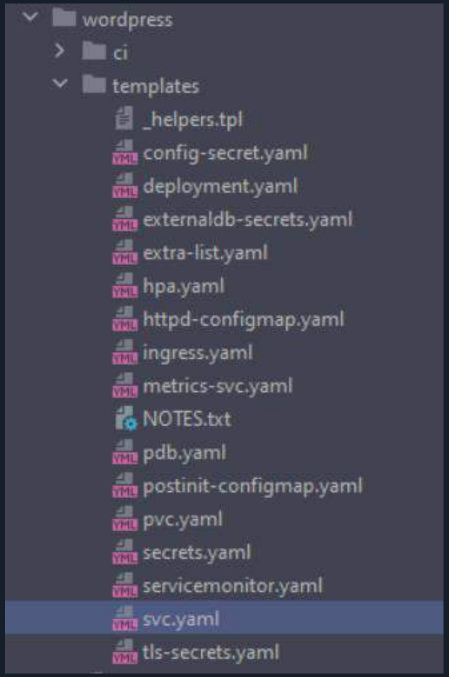
values.yaml

```
replicaCount: 1

image:
  repository: nginx
```

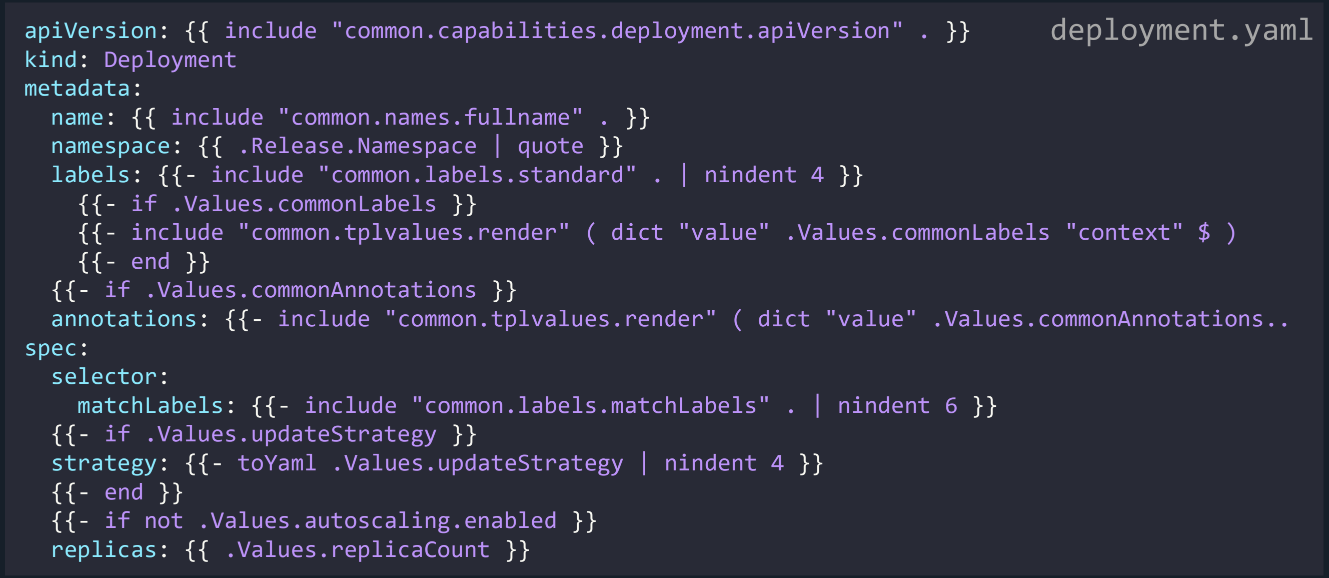
deployment.yaml

```
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
```



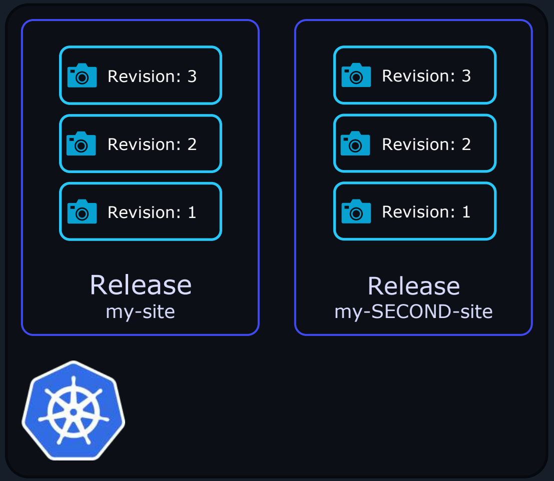
deployment.yaml

```
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 }}
```

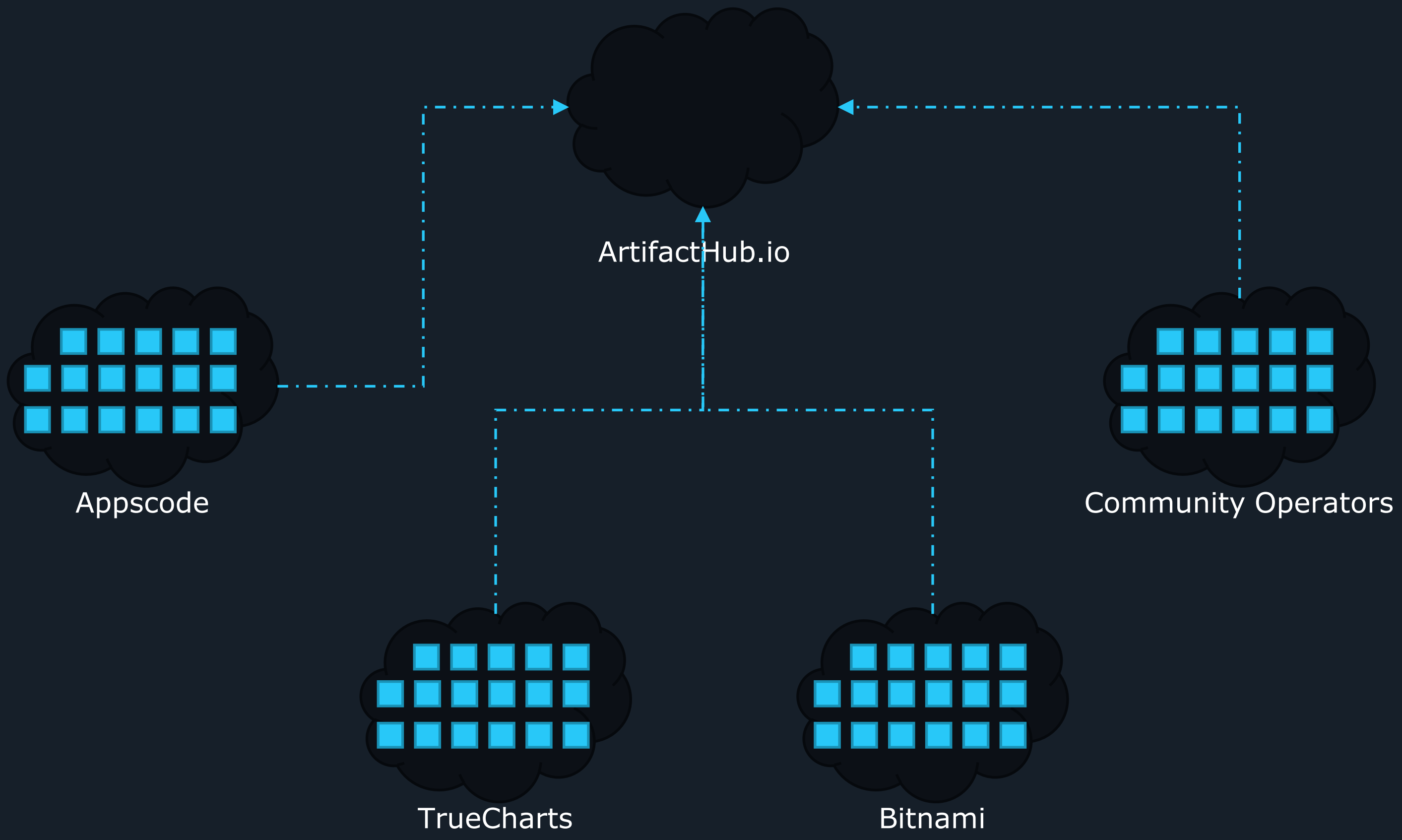
wordpress

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
```

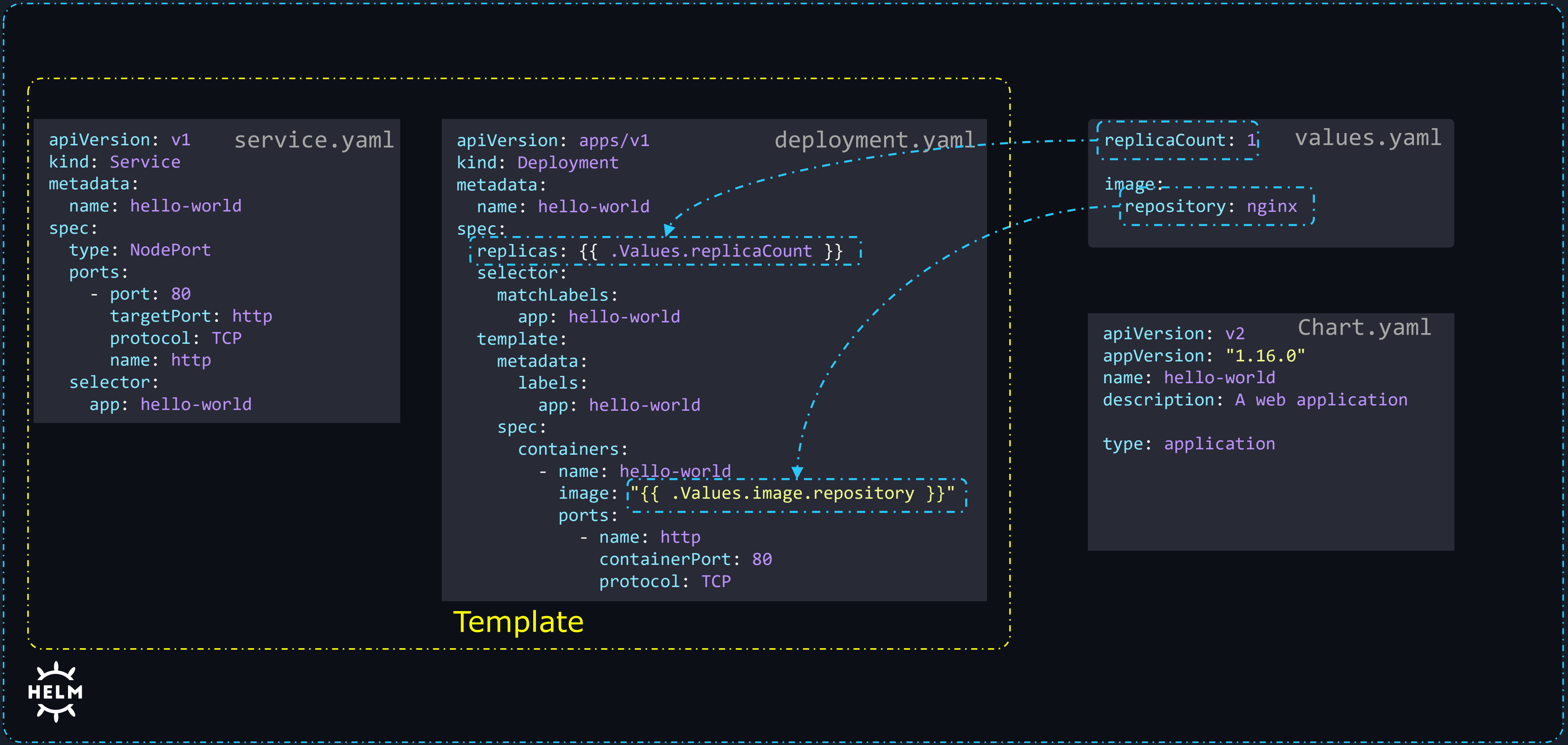


Helm Repositories



Helm Charts

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
```

Template

values.yaml

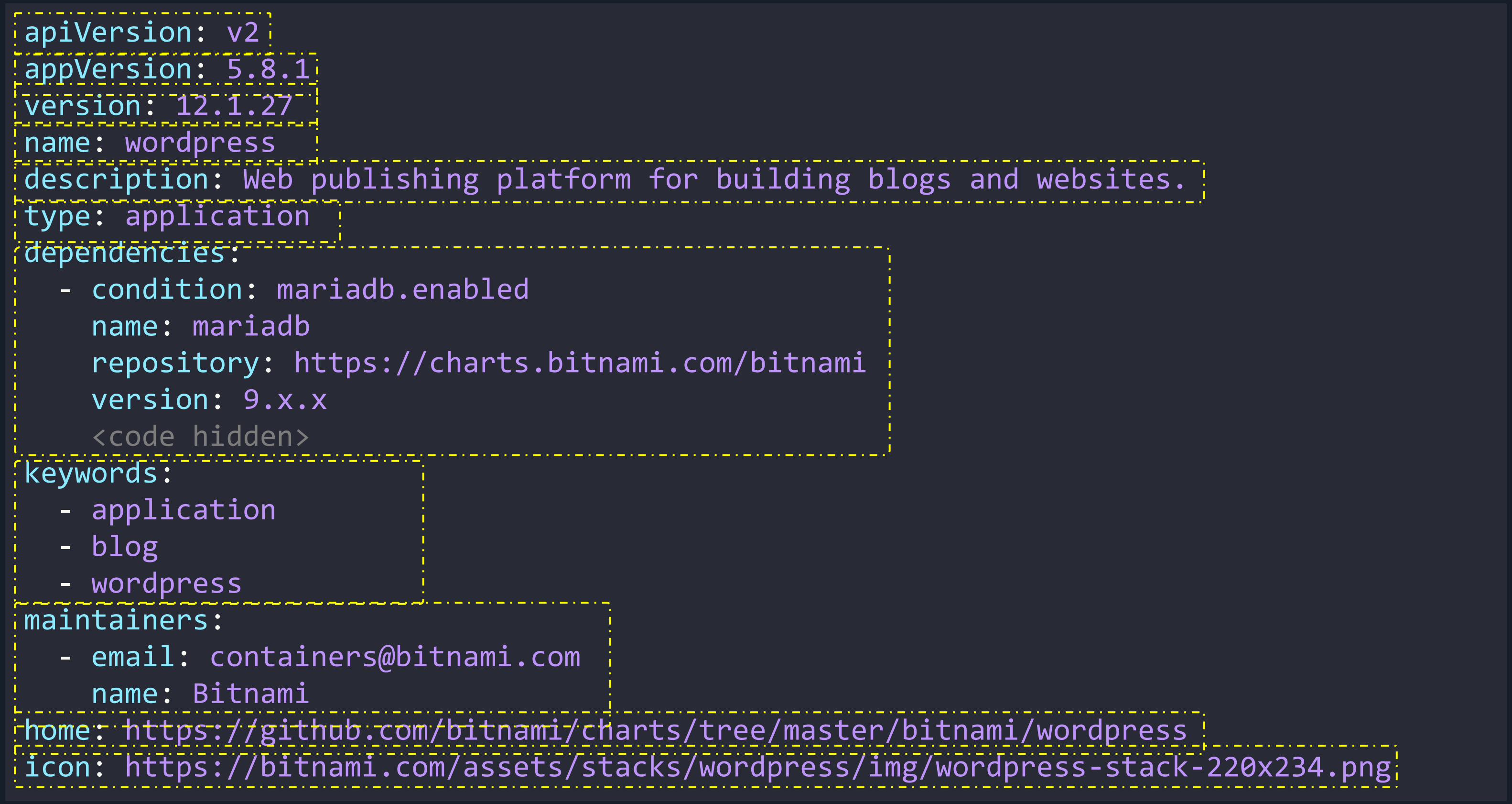
```yaml
replicaCount: 1

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
```

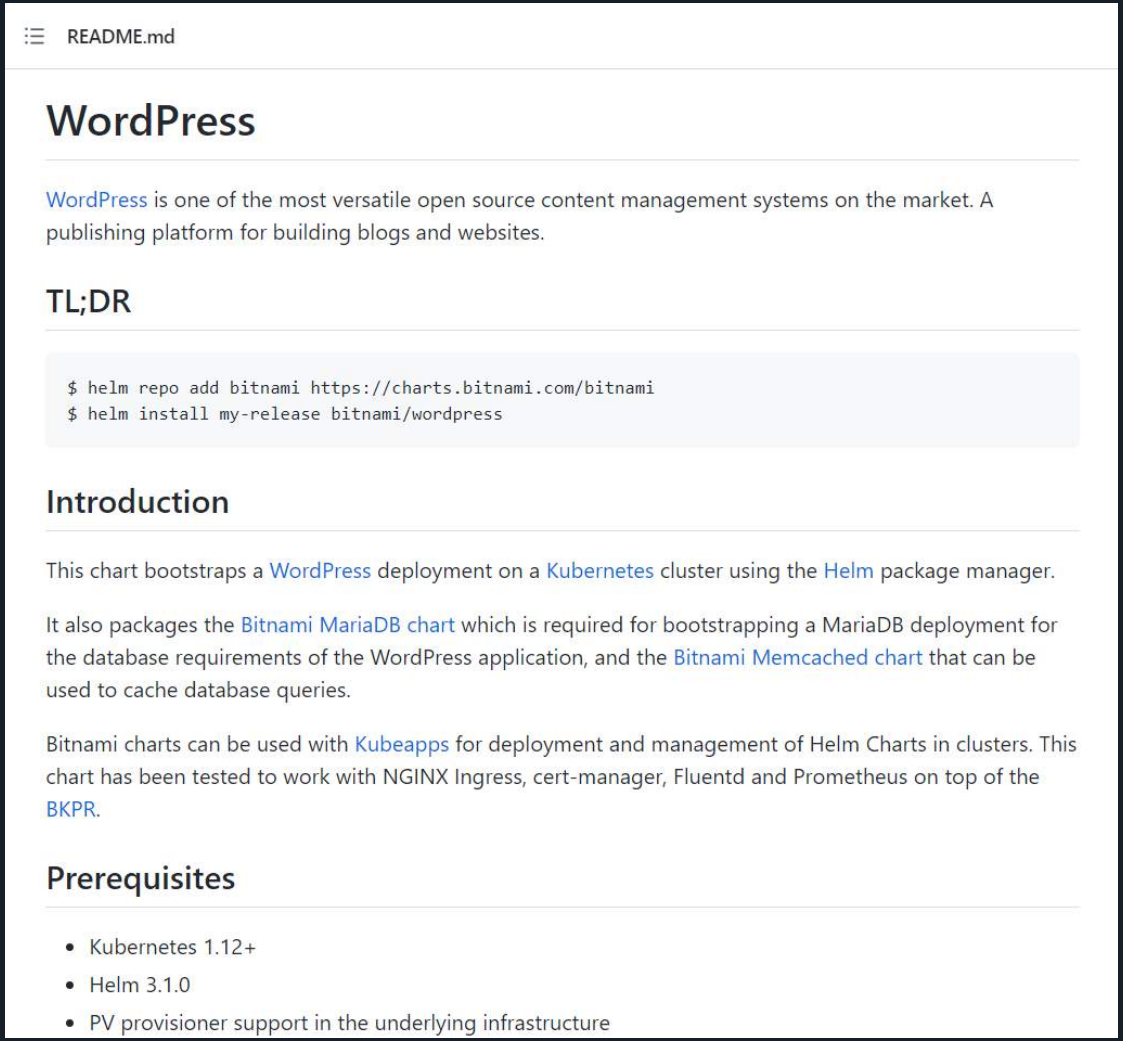
Helm 2	Helm 3
v1	v2

Types

application	library

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.

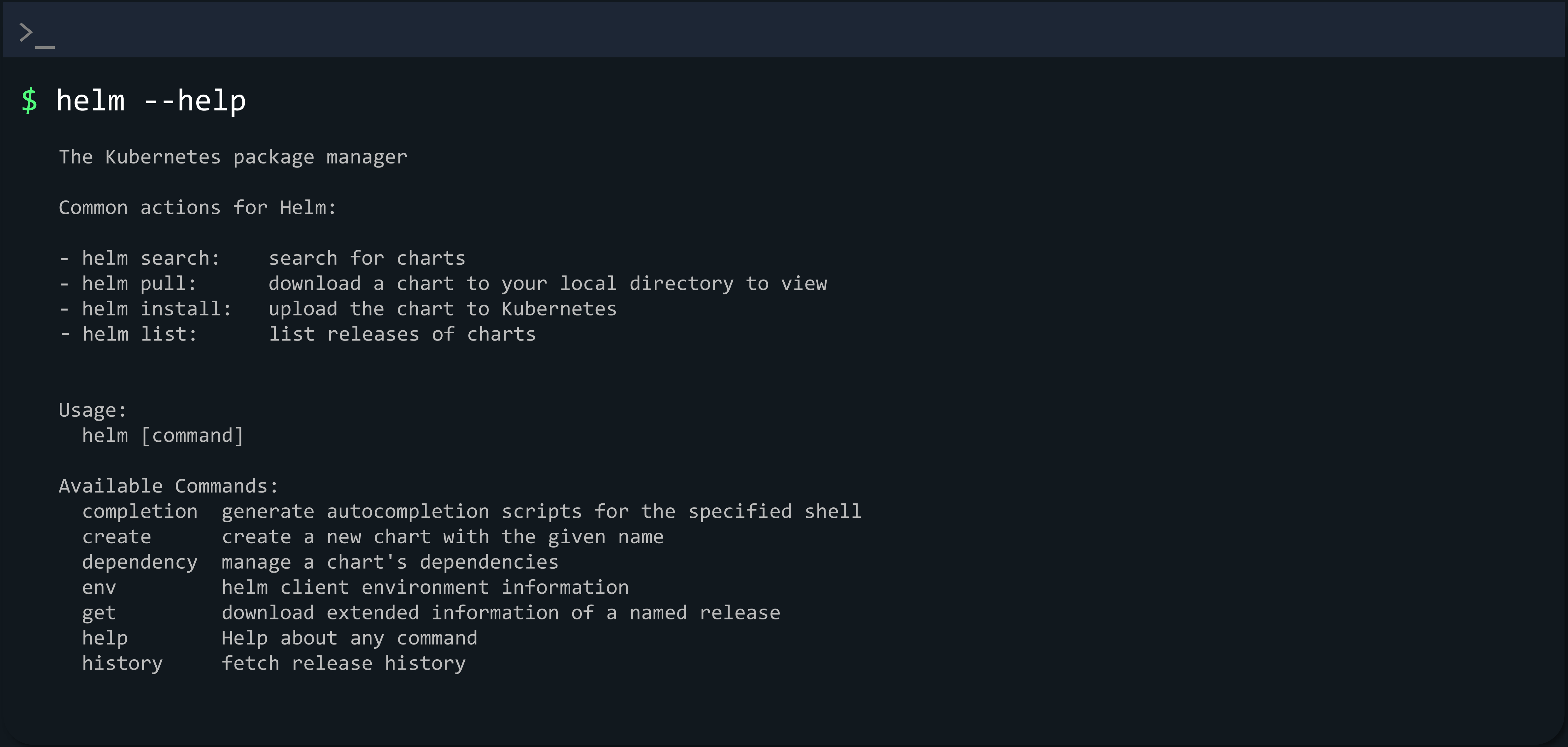
Bitnami charts can be used with Kubeapps for deployment and management of Helm Charts in clusters. This chart has been tested to work with NGINX Ingress, cert-manager, Fluentd and Prometheus on top of the BKPR.

Prerequisites

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

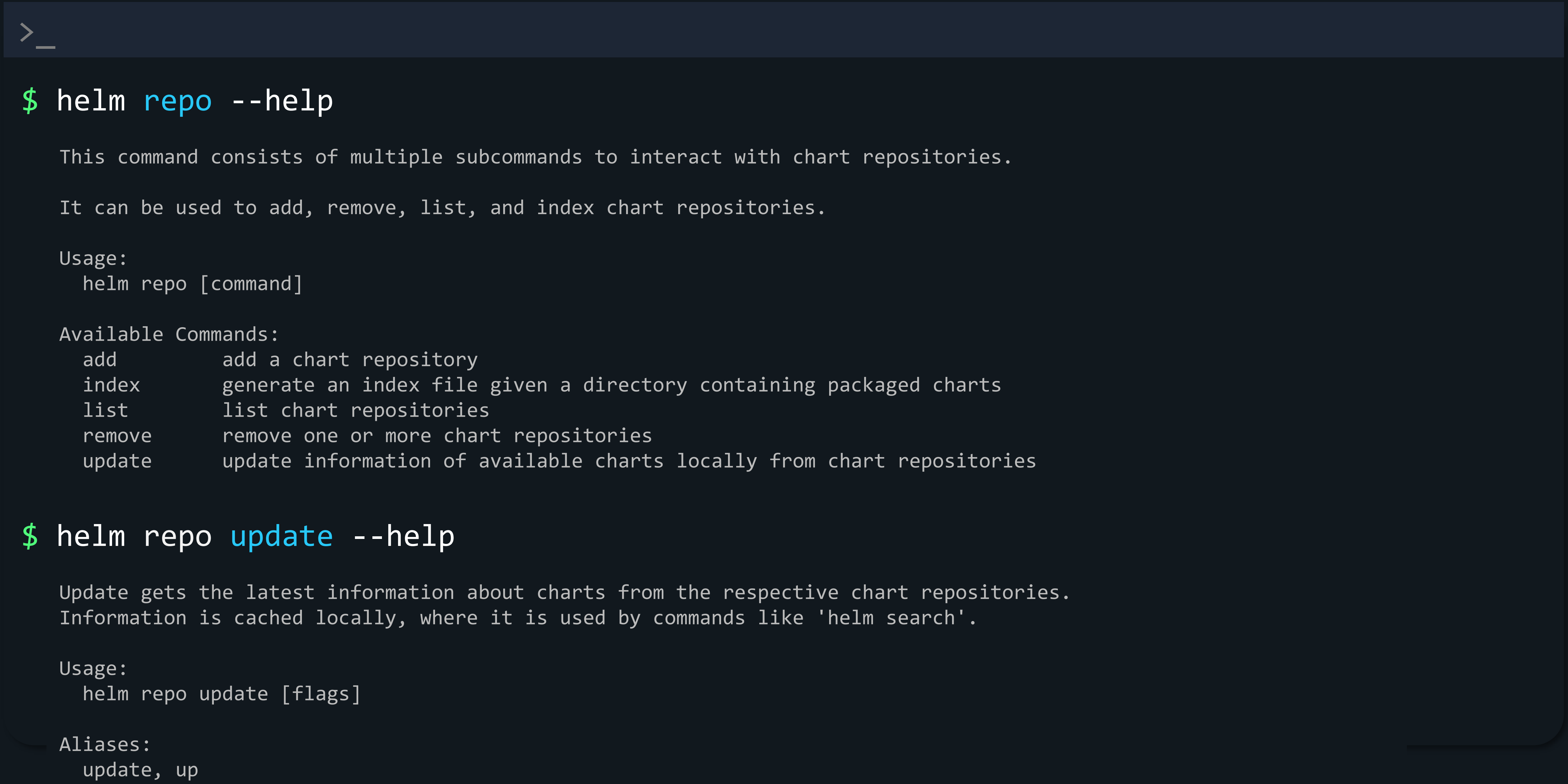
Working with Helm

Helm CLI



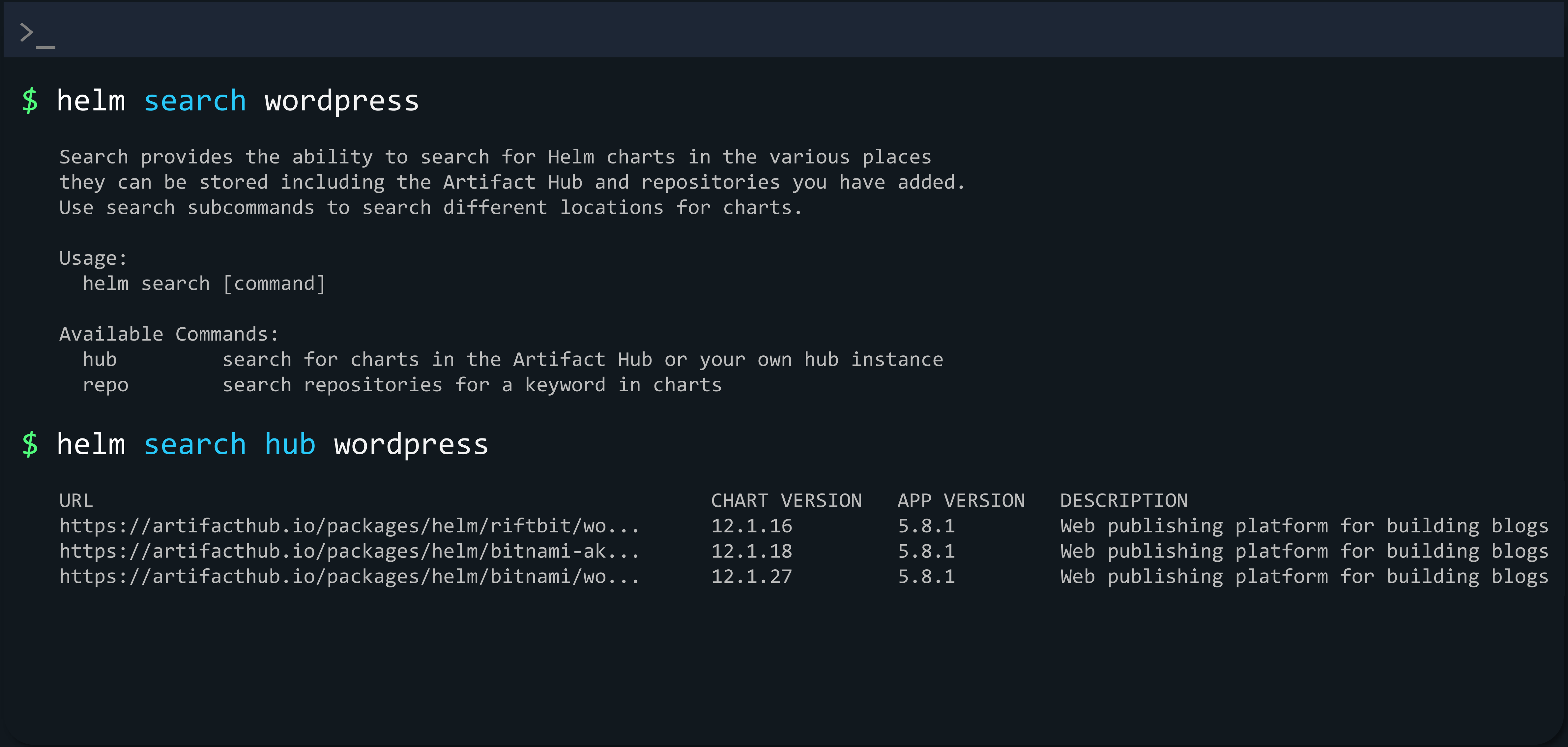
```
$ helm --help

   The Kubernetes package manager

   Common actions for Helm:

   - helm search:    search for charts
   - helm pull:      download a chart to your local directory to view
   - helm install:   upload the chart to Kubernetes
   - helm list:      list releases of charts


   Usage:
     helm [command]

   Available Commands:
     completion  generate autocompletion scripts for the specified shell
     create      create a new chart with the given name
     dependency  manage a chart's dependencies
     env         helm client environment information
     get         download extended information of a named release
     help        Help about any command
     history     fetch release history
```

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
```

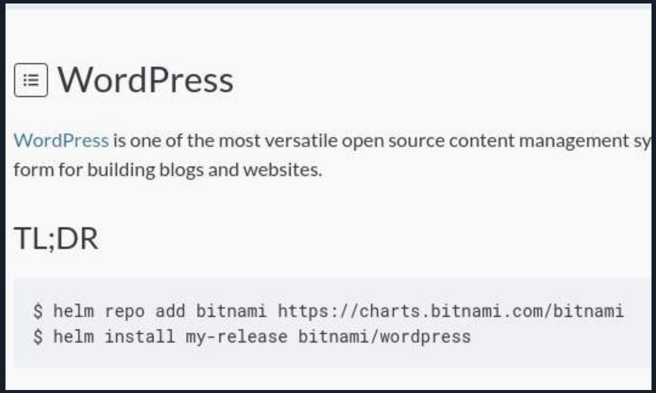
Wordpress



```
$ 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)
```

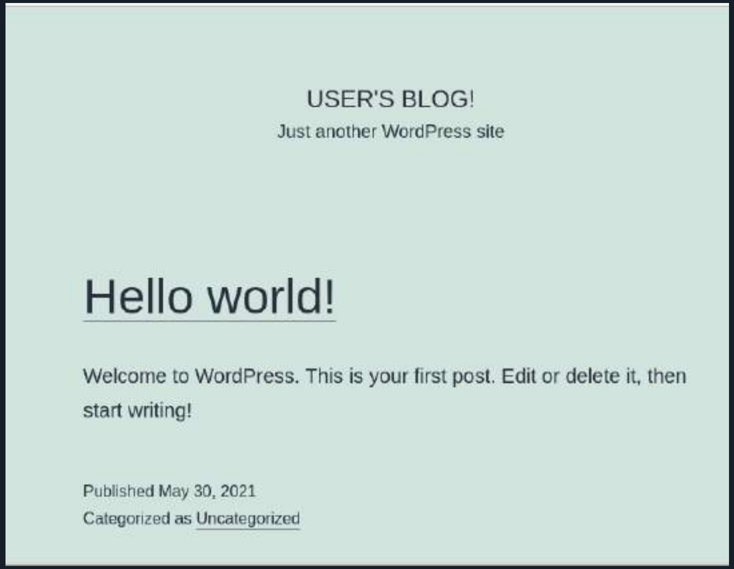


WordPress

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!

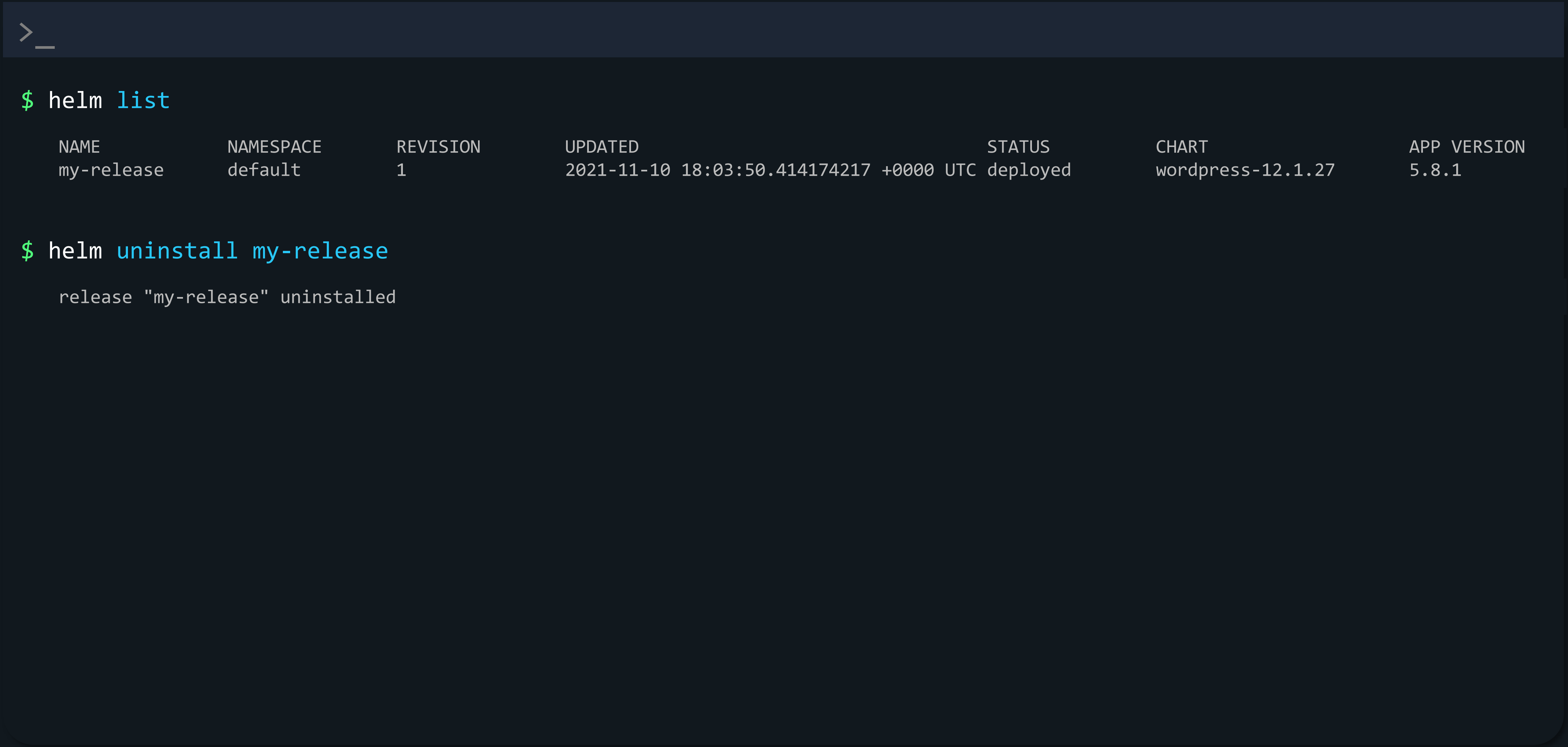
Just another WordPress site

Hello world!

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

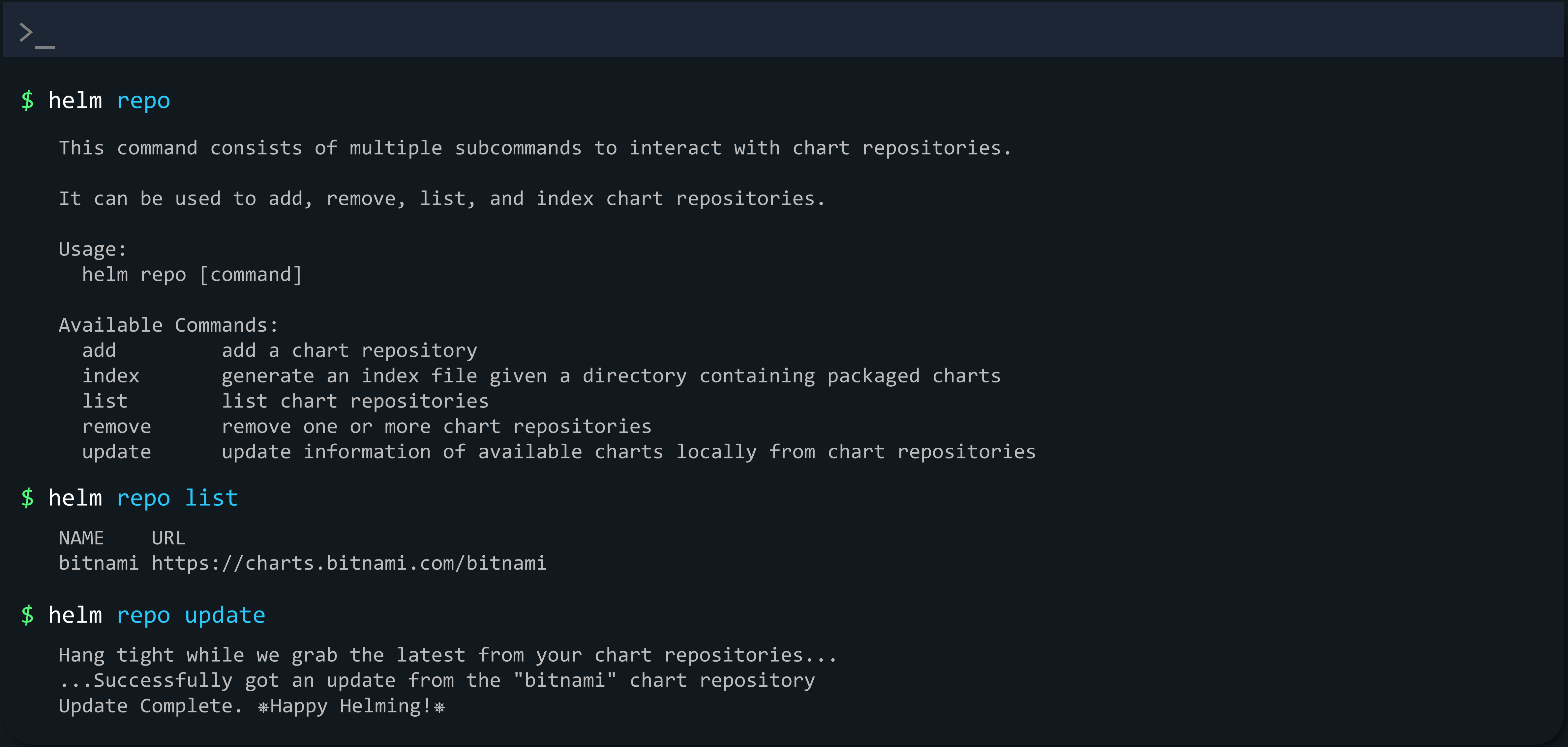
Published May 30, 2021
Categorized as Uncategorized

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
```

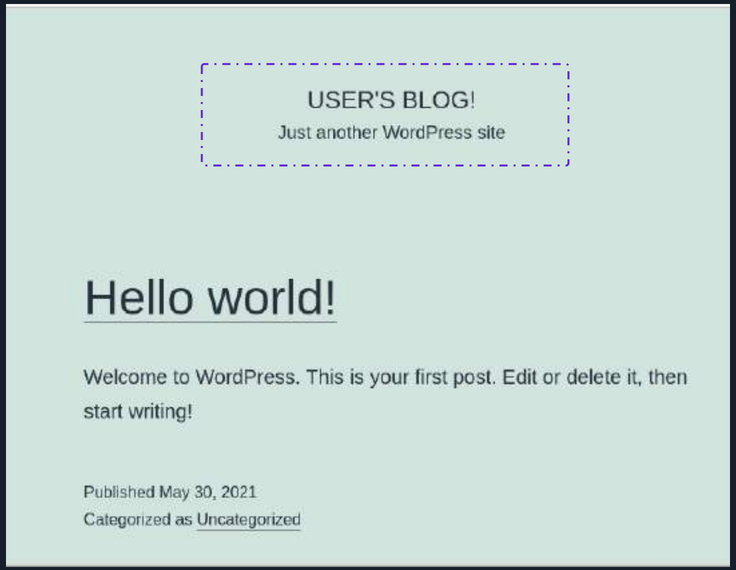
Helm Repo



```
$ 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

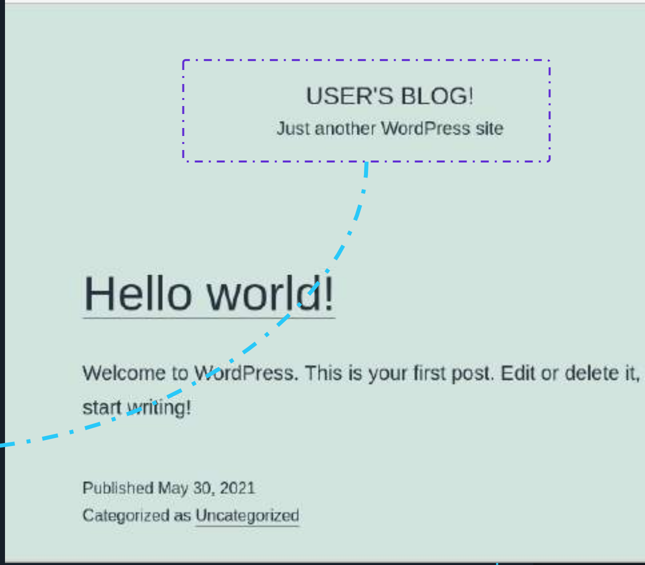
Helm Charts

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!

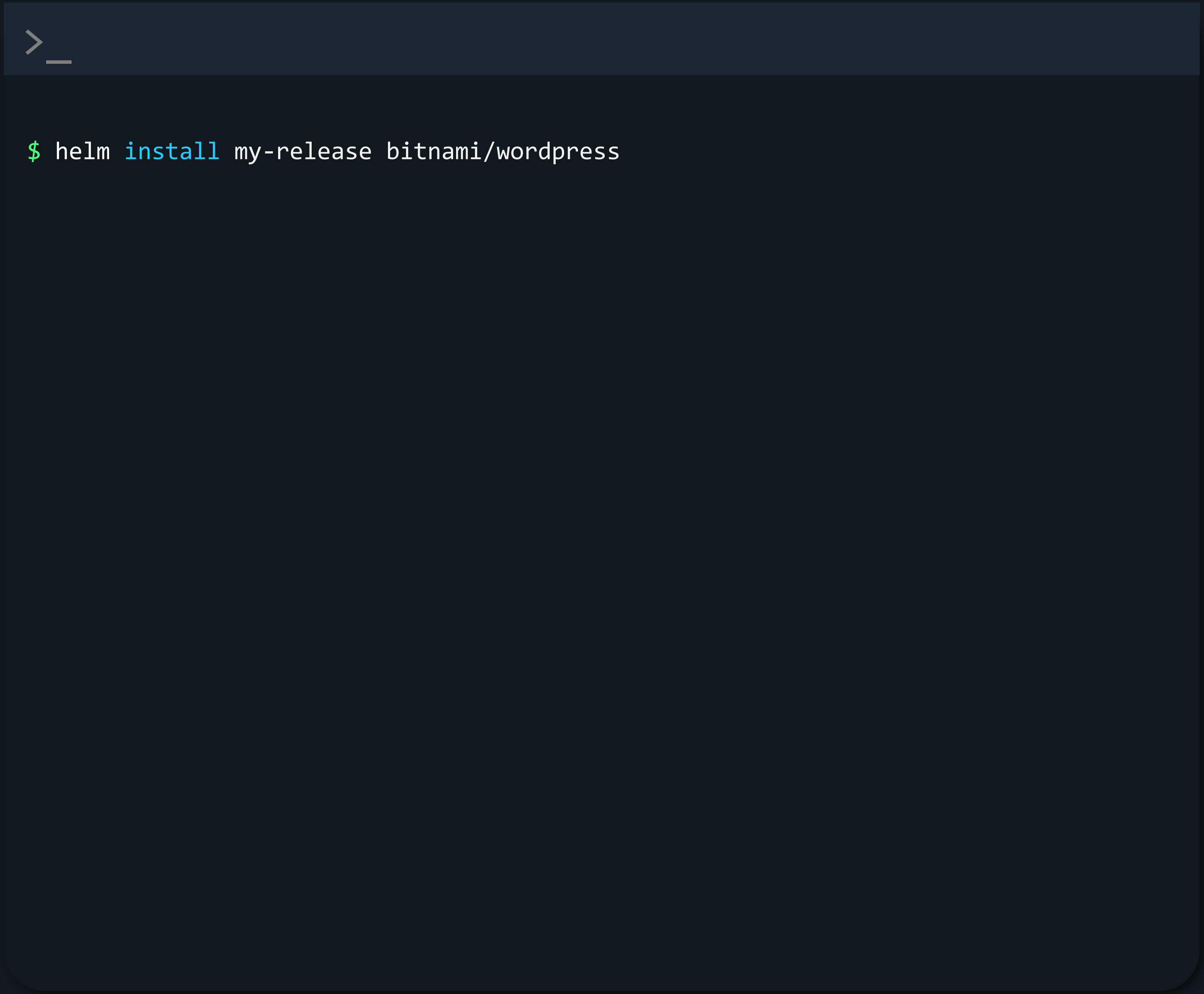
Just another WordPress site

Hello world!

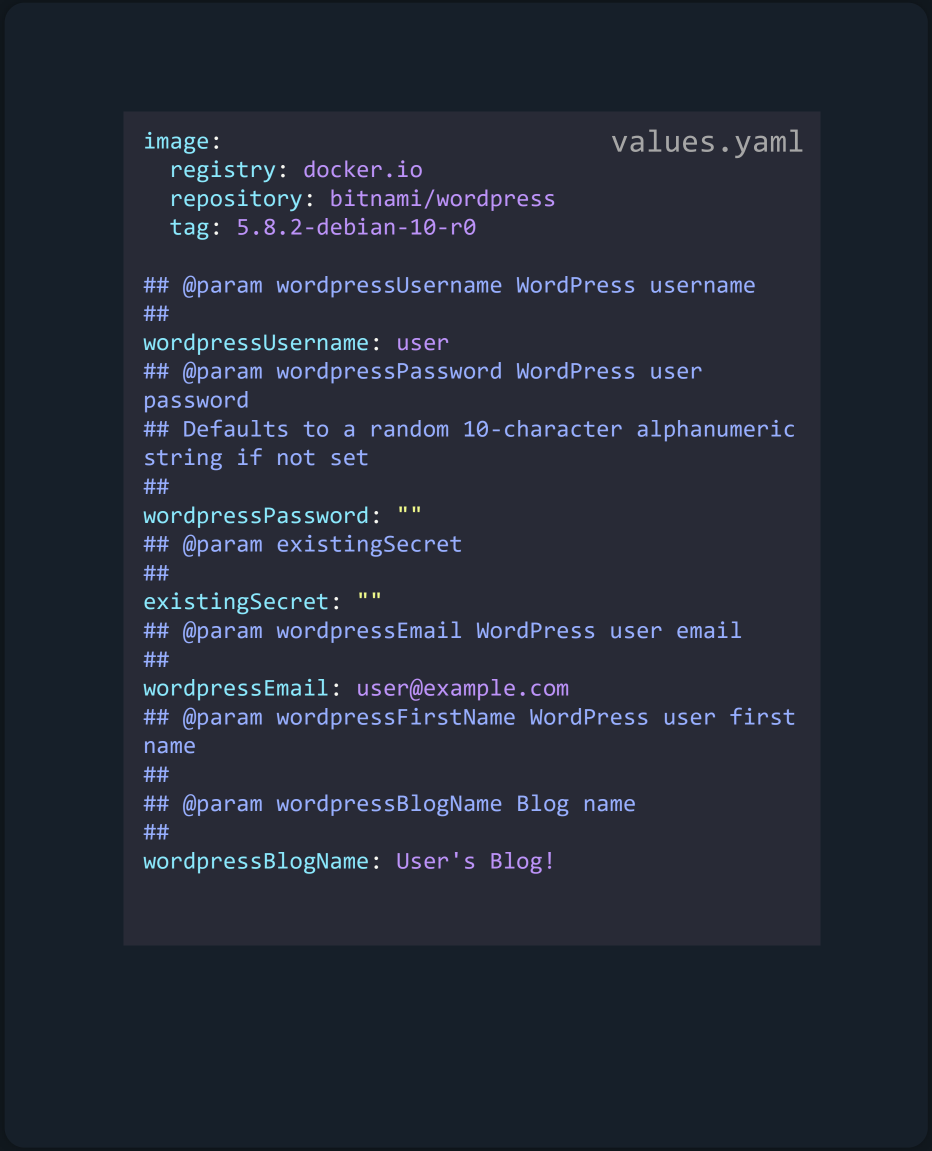
Welcome to WordPress. This is your first post. Edit or delete it, start writing!

Published May 30, 2021

Categorized as Uncategorized



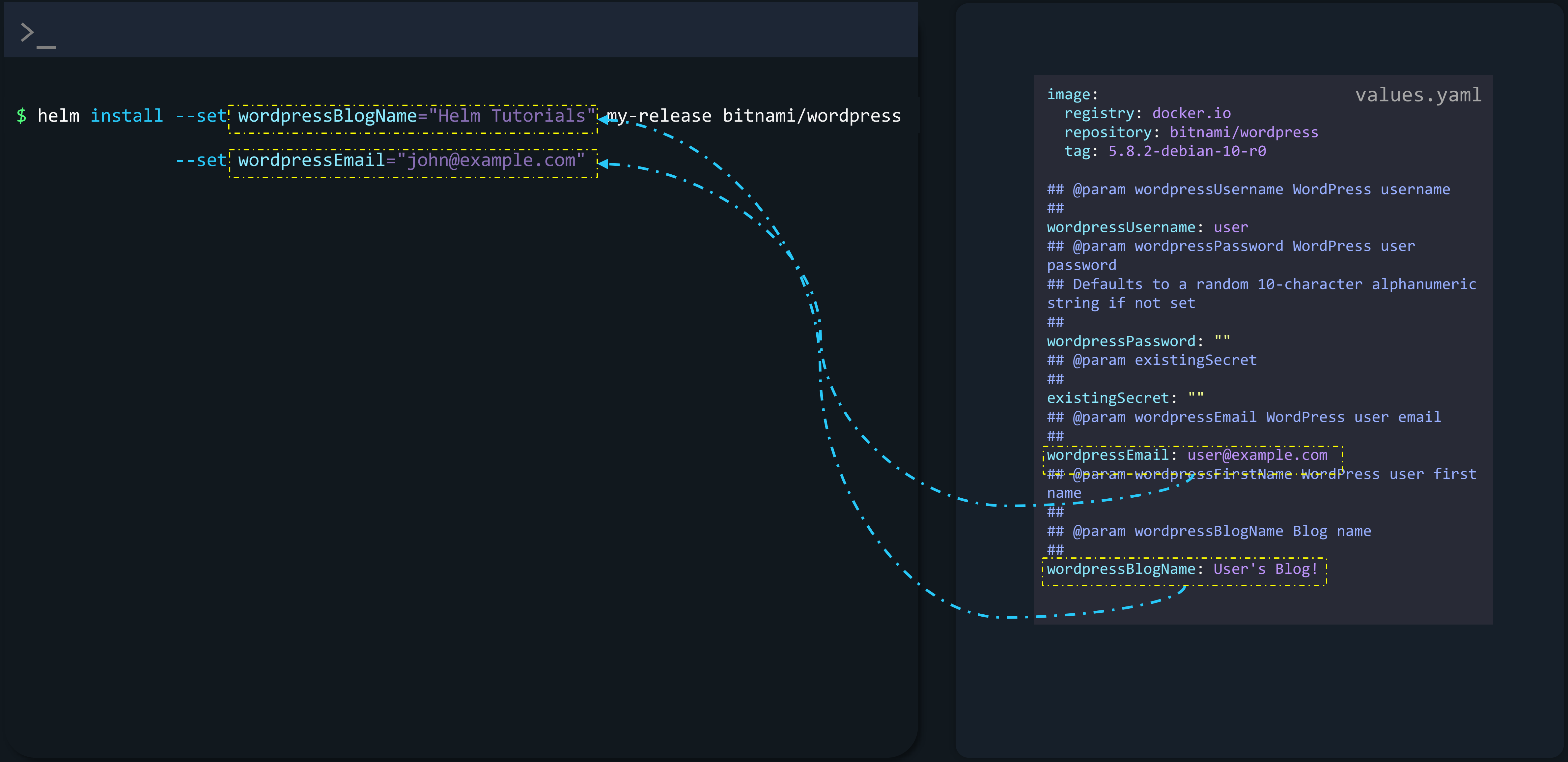
```
$ helm install my-release bitnami/wordpress
```



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

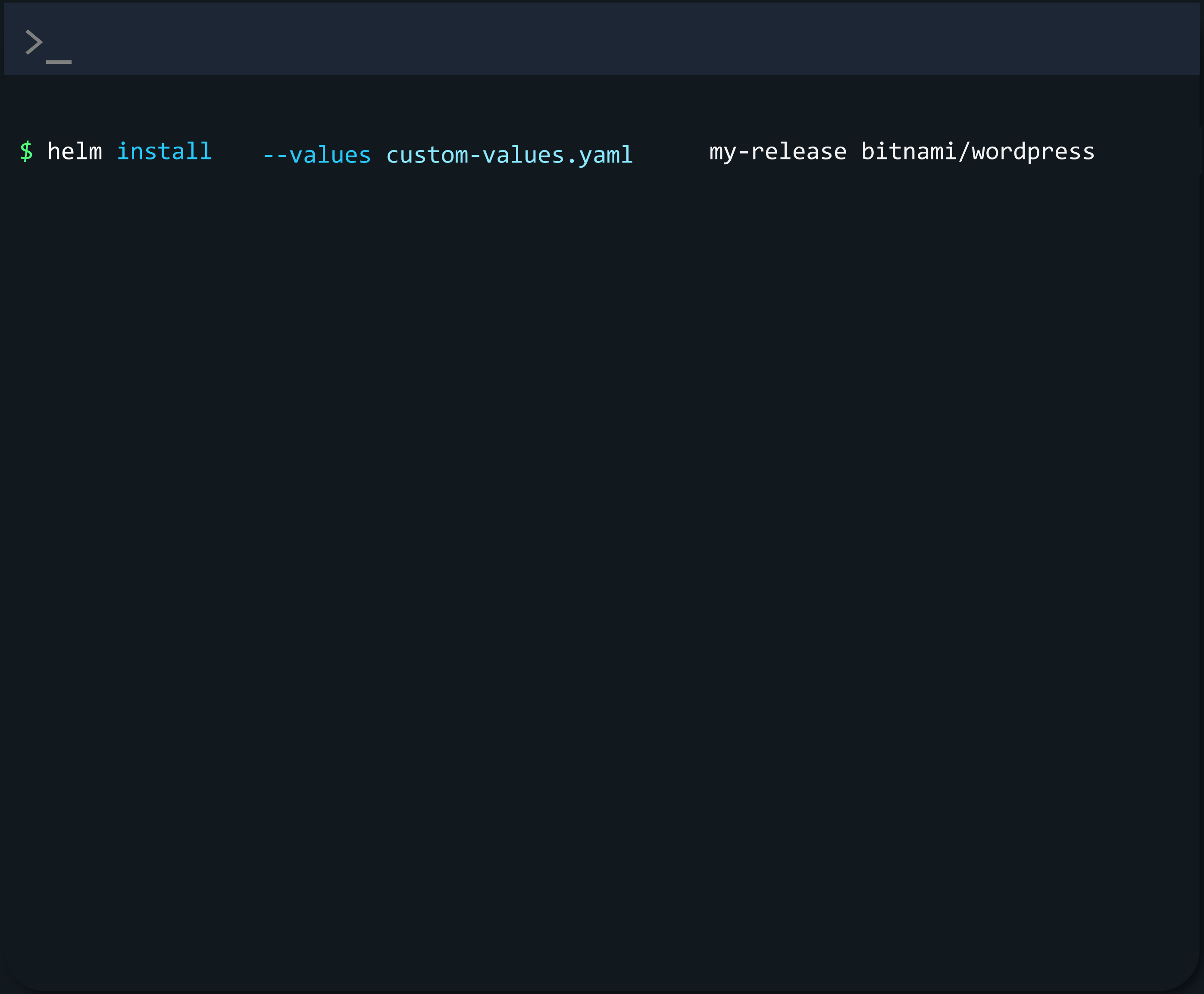


```
$ helm install --set wordpressBlogName="Helm Tutorials" my-release bitnami/wordpress

               --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

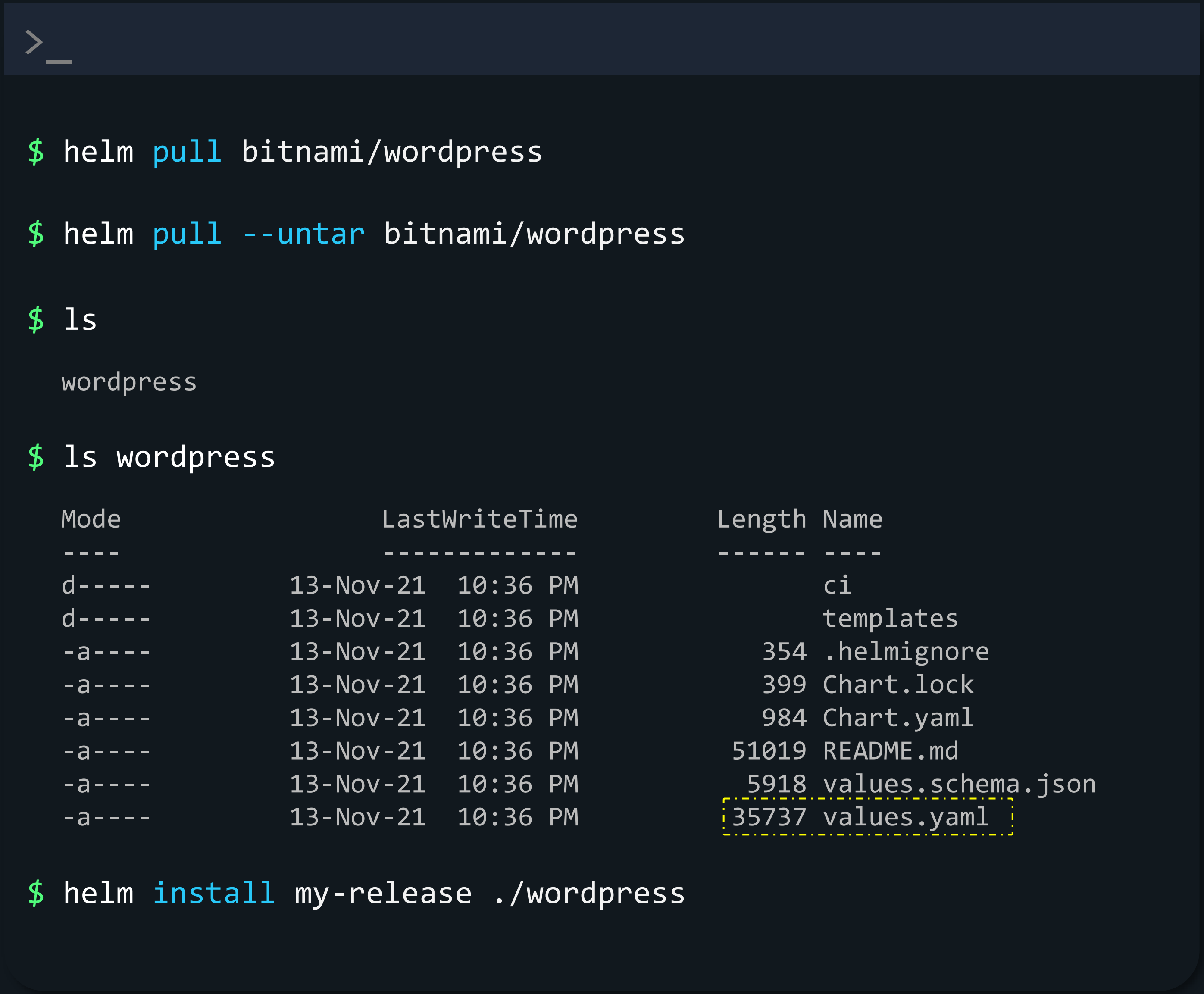


```
$ helm install    --values custom-values.yaml     my-release bitnami/wordpress
```

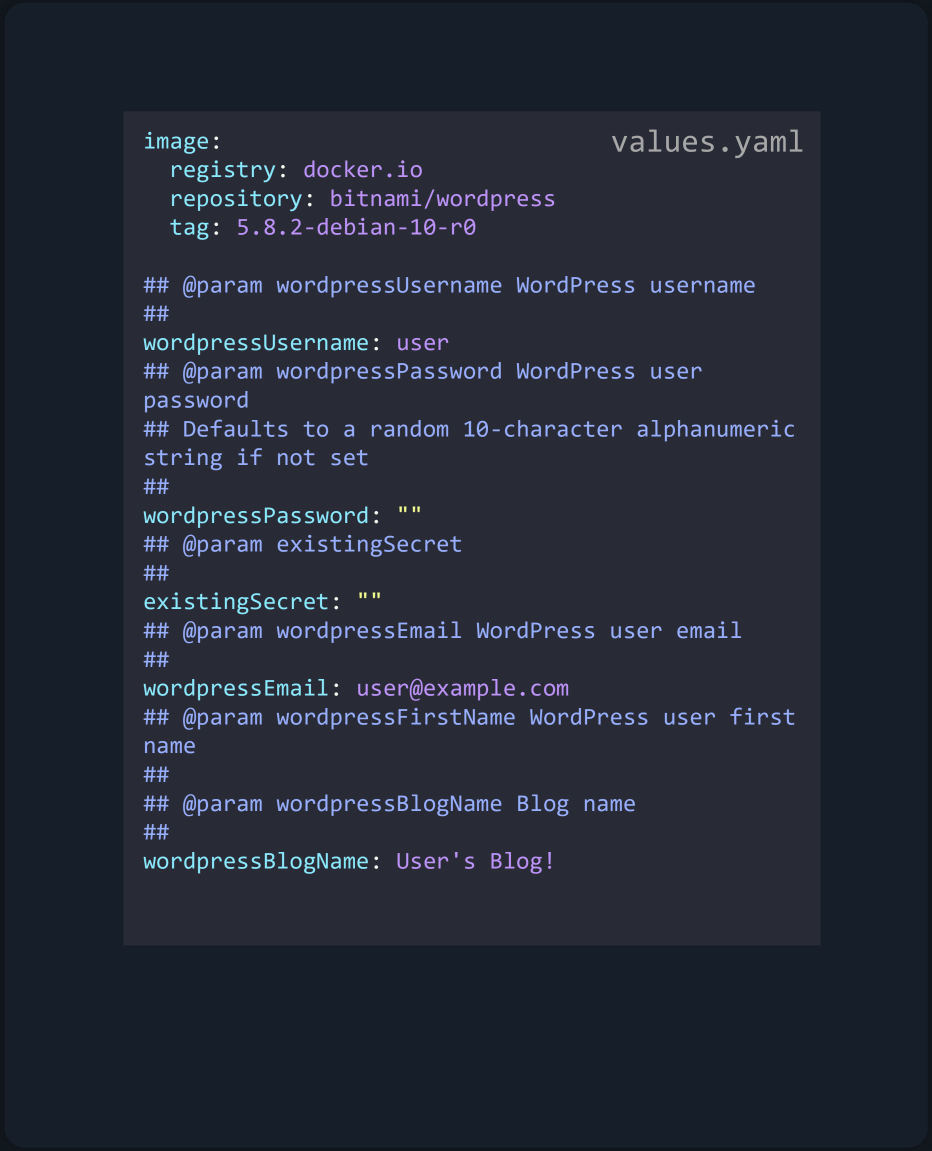
custom-values.yaml

```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
```



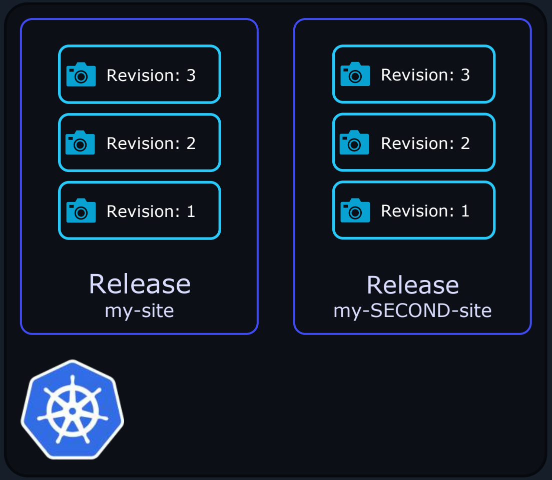
values.yaml

```
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!
```

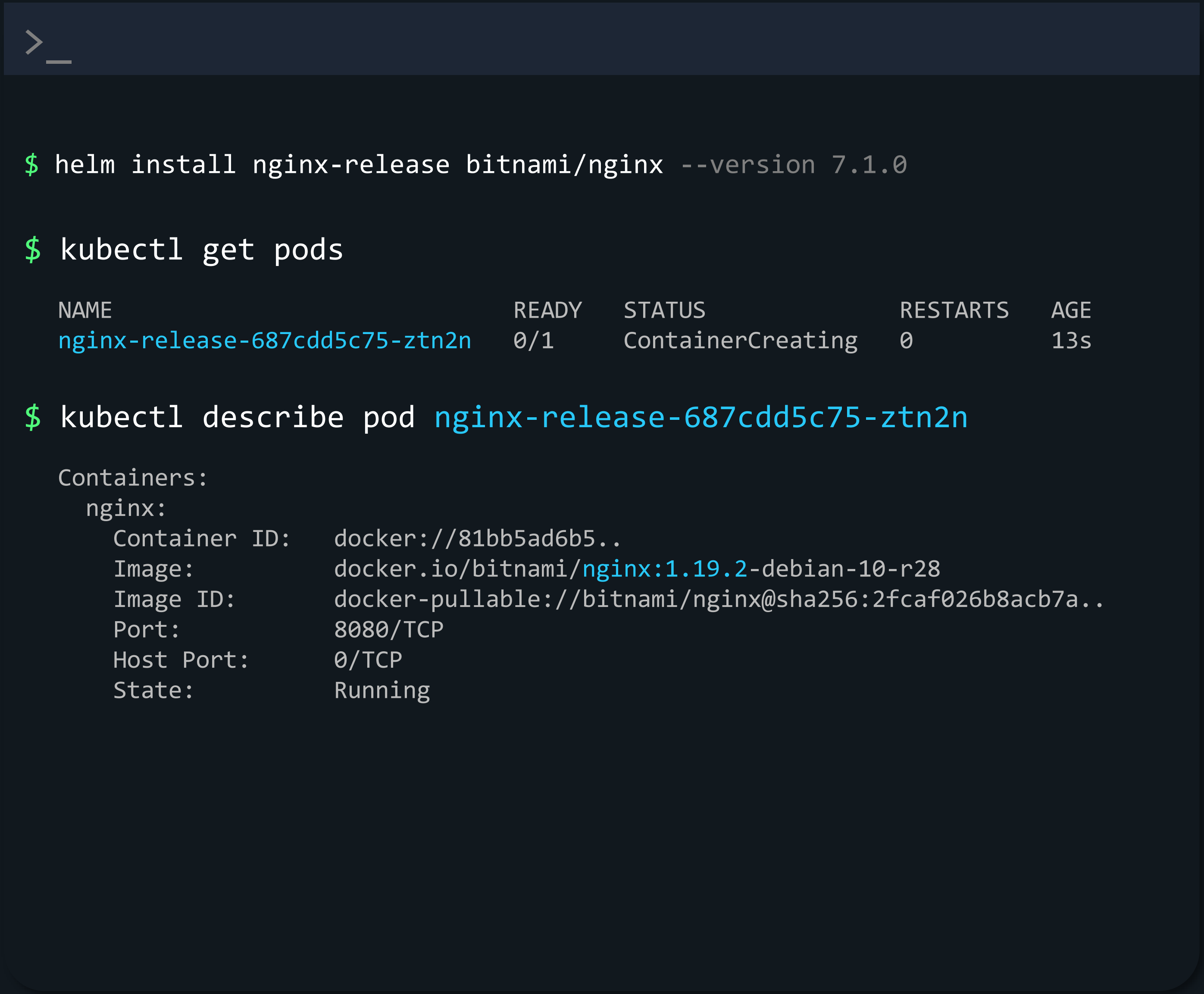
Lifecycle Management

Helm Releases

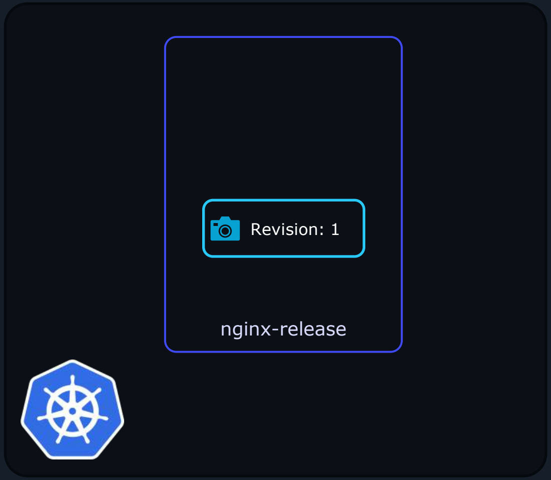
```
>_

$ helm install my-site bitnami/wordpress

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



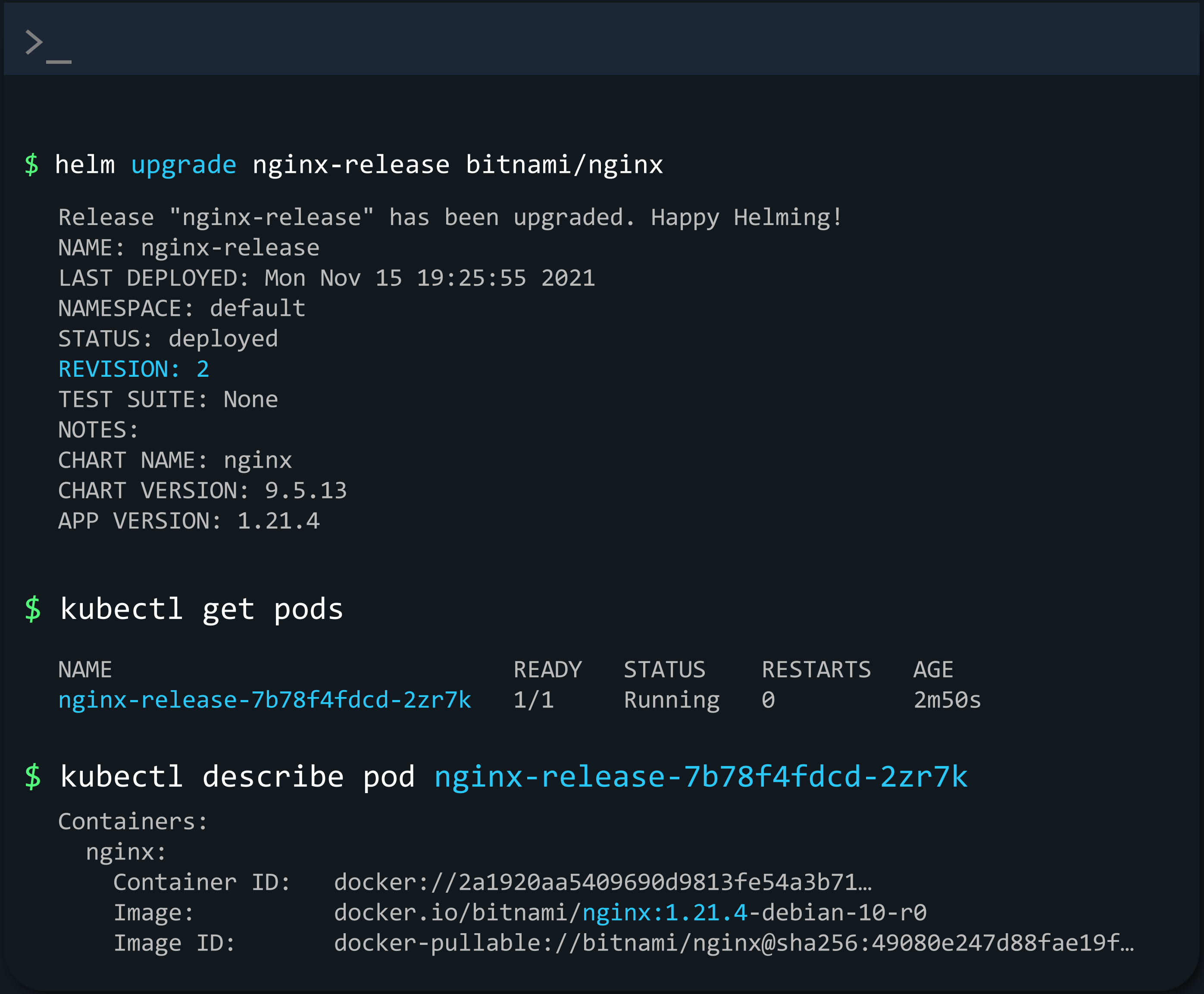
Helm Upgrade



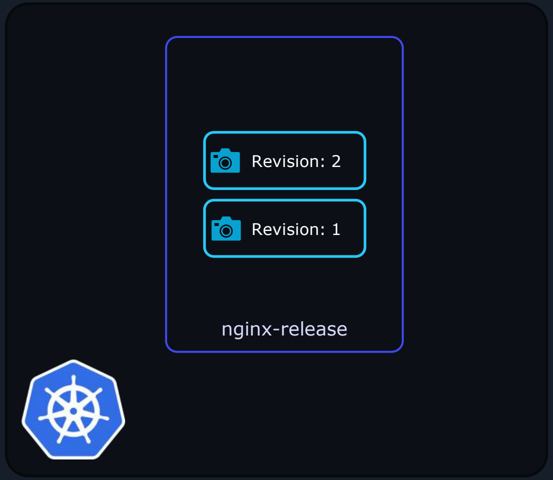
```
$ helm install nginx-release bitnami/nginx --version 7.1.0

$ 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
```



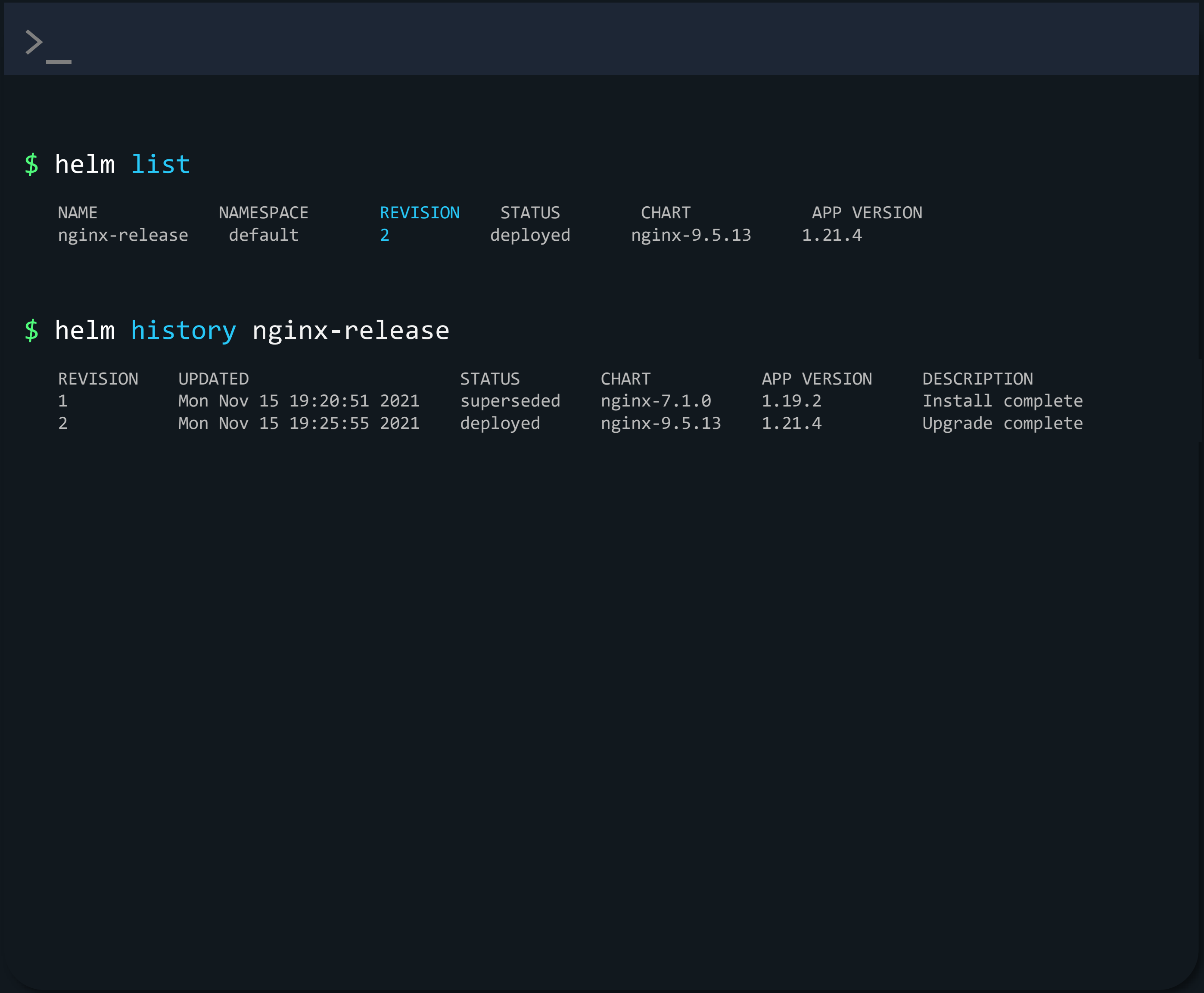
Helm Upgrade



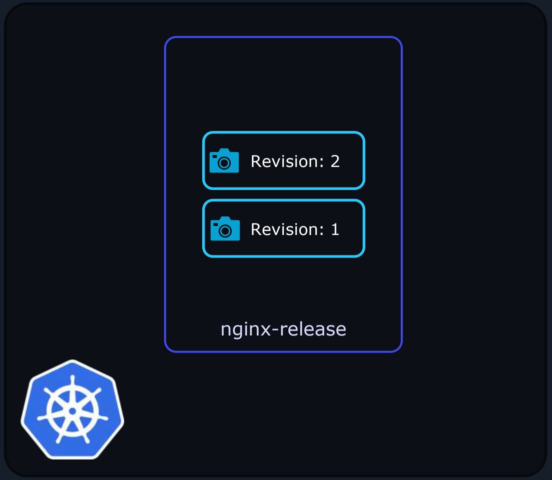
```
$ helm upgrade nginx-release bitnami/nginx

  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…
```



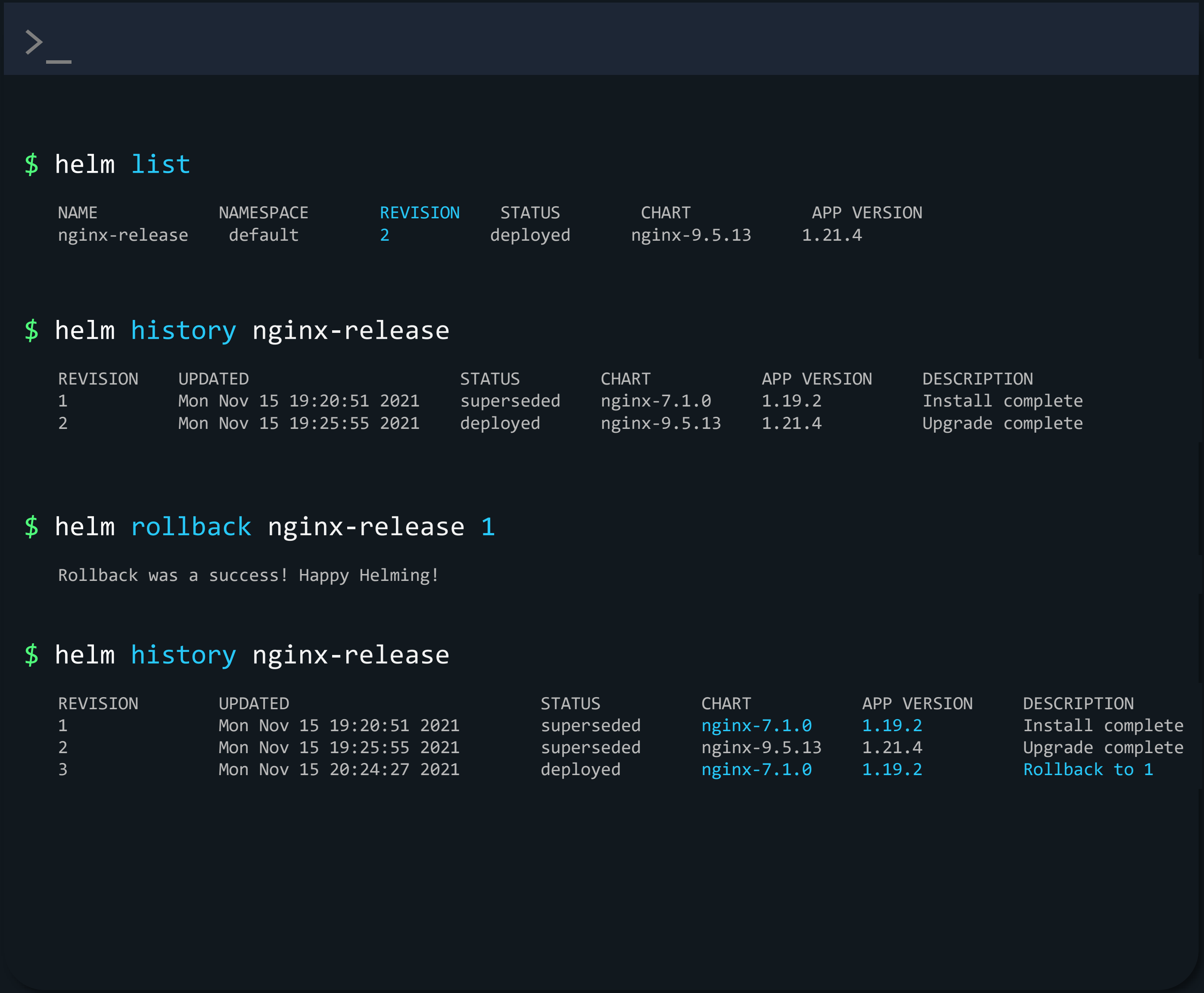
Helm Upgrade



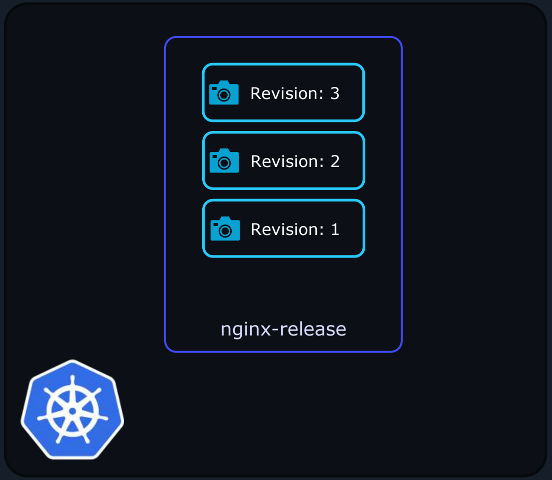
```
$ helm list

  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

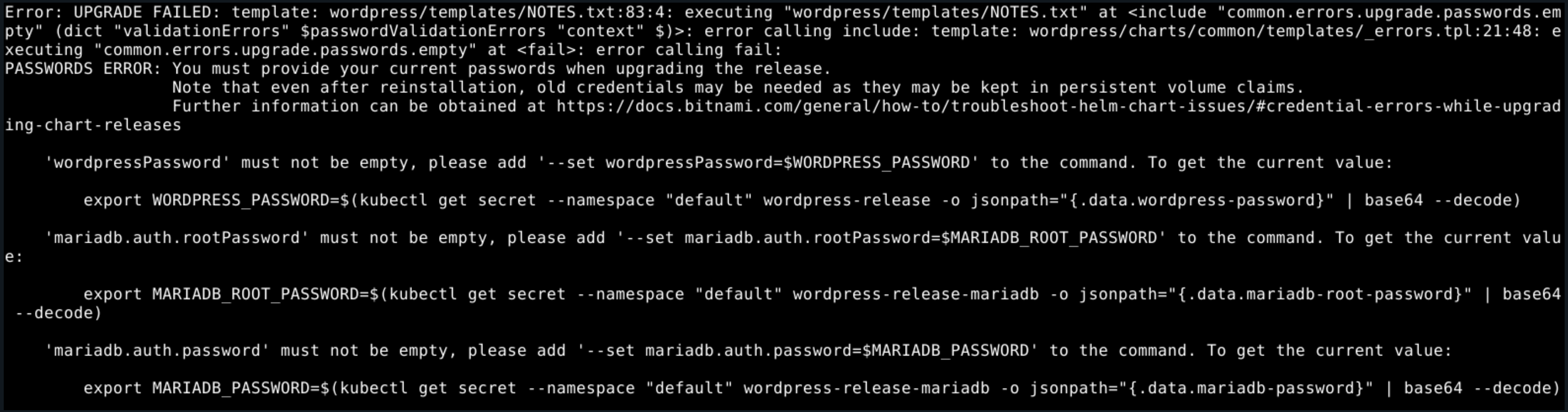


```
$ helm list

  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
```



Helm Upgrade

```
$ helm upgrade wordpress-release bitnami/wordpress
```

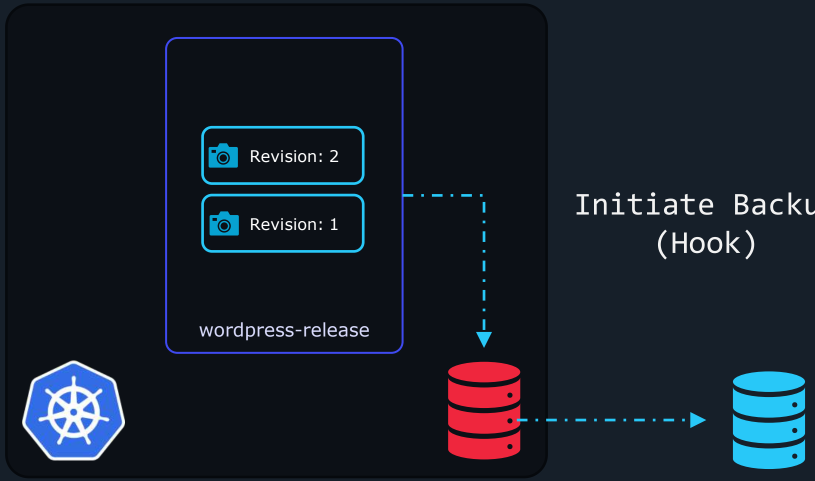


```
Error: UPGRADE FAILED: template: wordpress/templates/NOTES.txt:83:4: executing "wordpress/templates/NOTES.txt" at <include "common.errors.upgrade.passwords.em
pty" (dict "validationErrors" $passwordValidationErrors "context" $)>: error calling include: template: wordpress/charts/common/templates/_errors.tpl:21:48: e
xecuting "common.errors.upgrade.passwords.empty" at <fail>: error calling fail:
PASSWORDS ERROR: You must provide your current passwords when upgrading the release.
                 Note that even after reinstallation, old credentials may be needed as they may be kept in persistent volume claims.
                 Further information can be obtained at https://docs.bitnami.com/general/how-to/troubleshoot-helm-chart-issues/#credential-errors-while-upgrad
ing-chart-releases

    'wordpressPassword' must not be empty, please add '--set wordpressPassword=$WORDPRESS_PASSWORD' to the command. To get the current value:

        export WORDPRESS_PASSWORD=$(kubectl get secret --namespace "default" wordpress-release -o jsonpath="{.data.wordpress-password}" | base64 --decode)

    'mariadb.auth.rootPassword' must not be empty, please add '--set mariadb.auth.rootPassword=$MARIADB_ROOT_PASSWORD' to the command. To get the current valu
e:

        export MARIADB_ROOT_PASSWORD=$(kubectl get secret --namespace "default" wordpress-release-mariadb -o jsonpath="{.data.mariadb-root-password}" | base64
 --decode)

    'mariadb.auth.password' must not be empty, please add '--set mariadb.auth.password=$MARIADB_PASSWORD' to the command. To get the current value:

        export MARIADB_PASSWORD=$(kubectl get secret --namespace "default" wordpress-release-mariadb -o jsonpath="{.data.mariadb-password}" | base64 --decode)
```

Writing a

Helm Charts

```
$ helm upgrade wordpress-release bitnami/wordpress
```



Initiate Backup
(Hook)

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
```

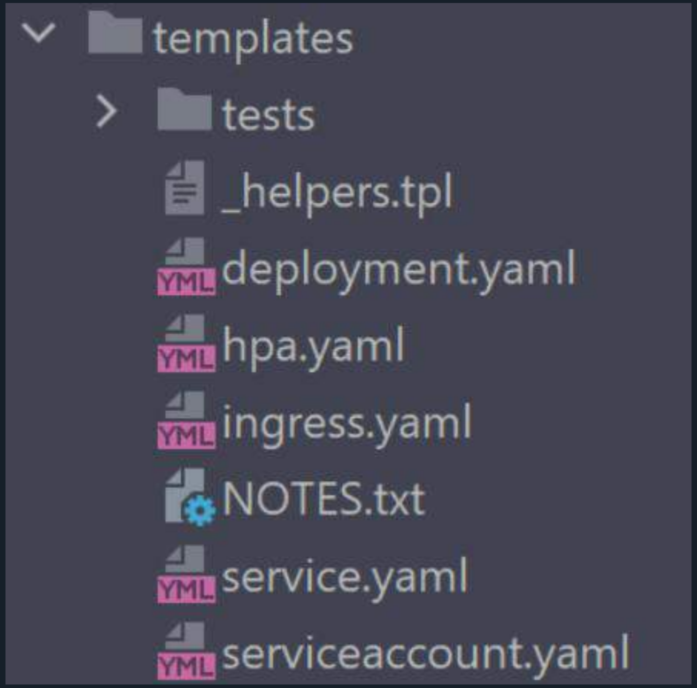
service.yaml

```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
```

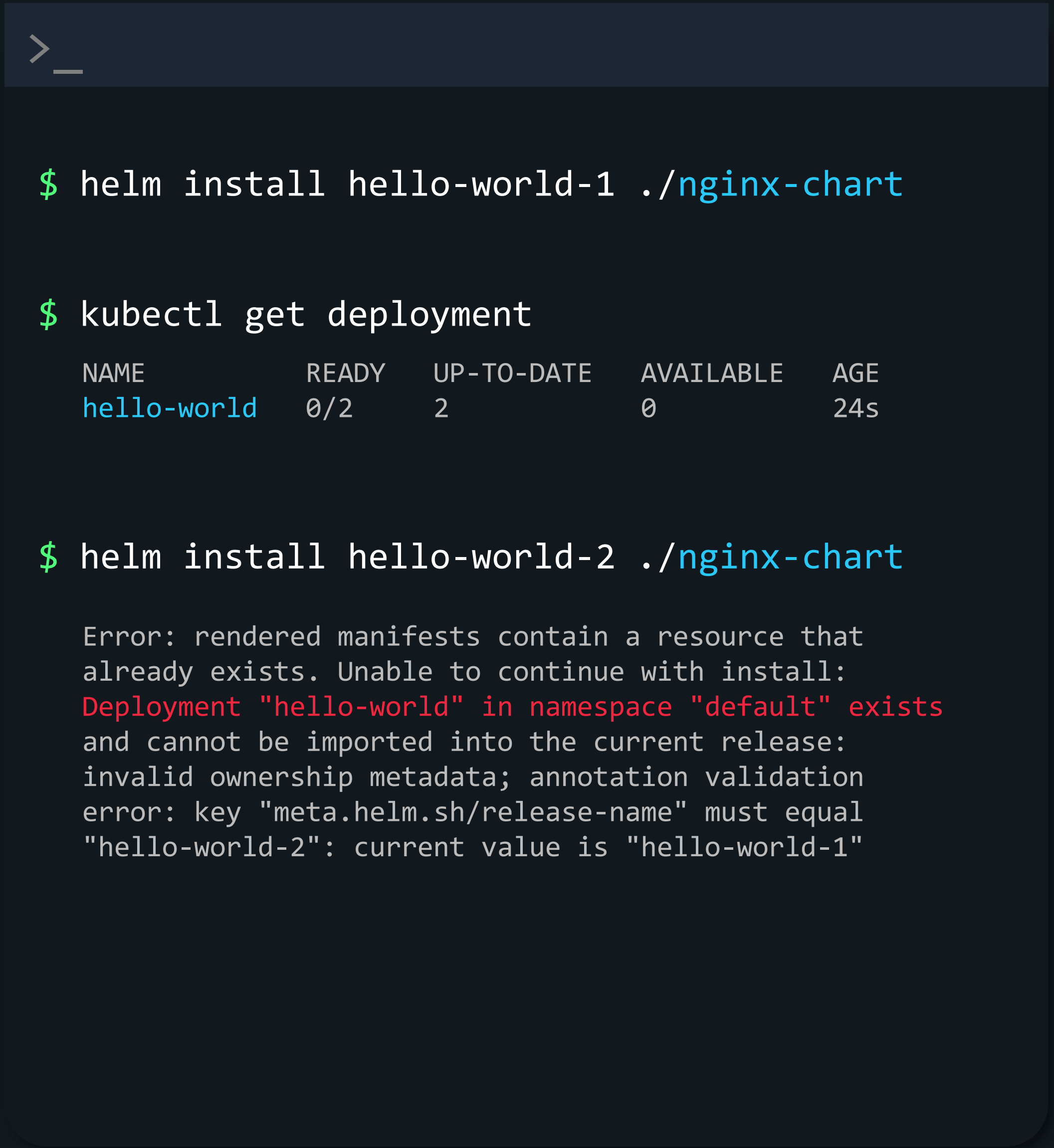
deployment.yaml

```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
```

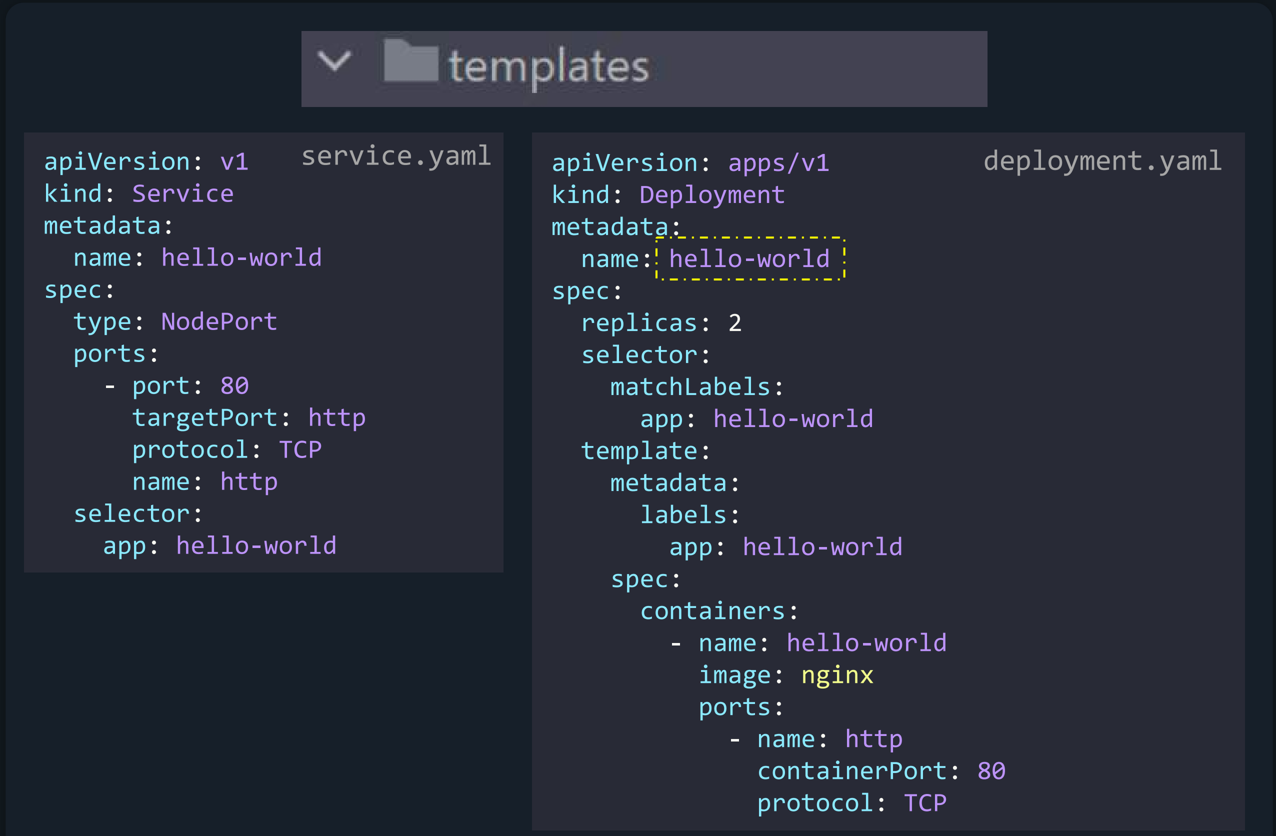
```
>_

$ 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"
```



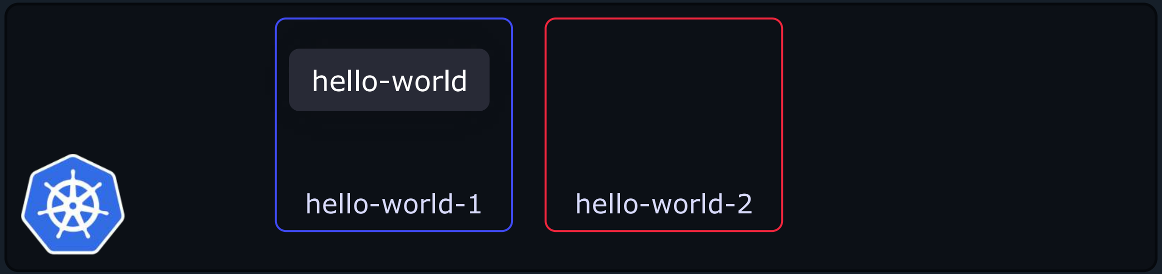
templates

service.yaml

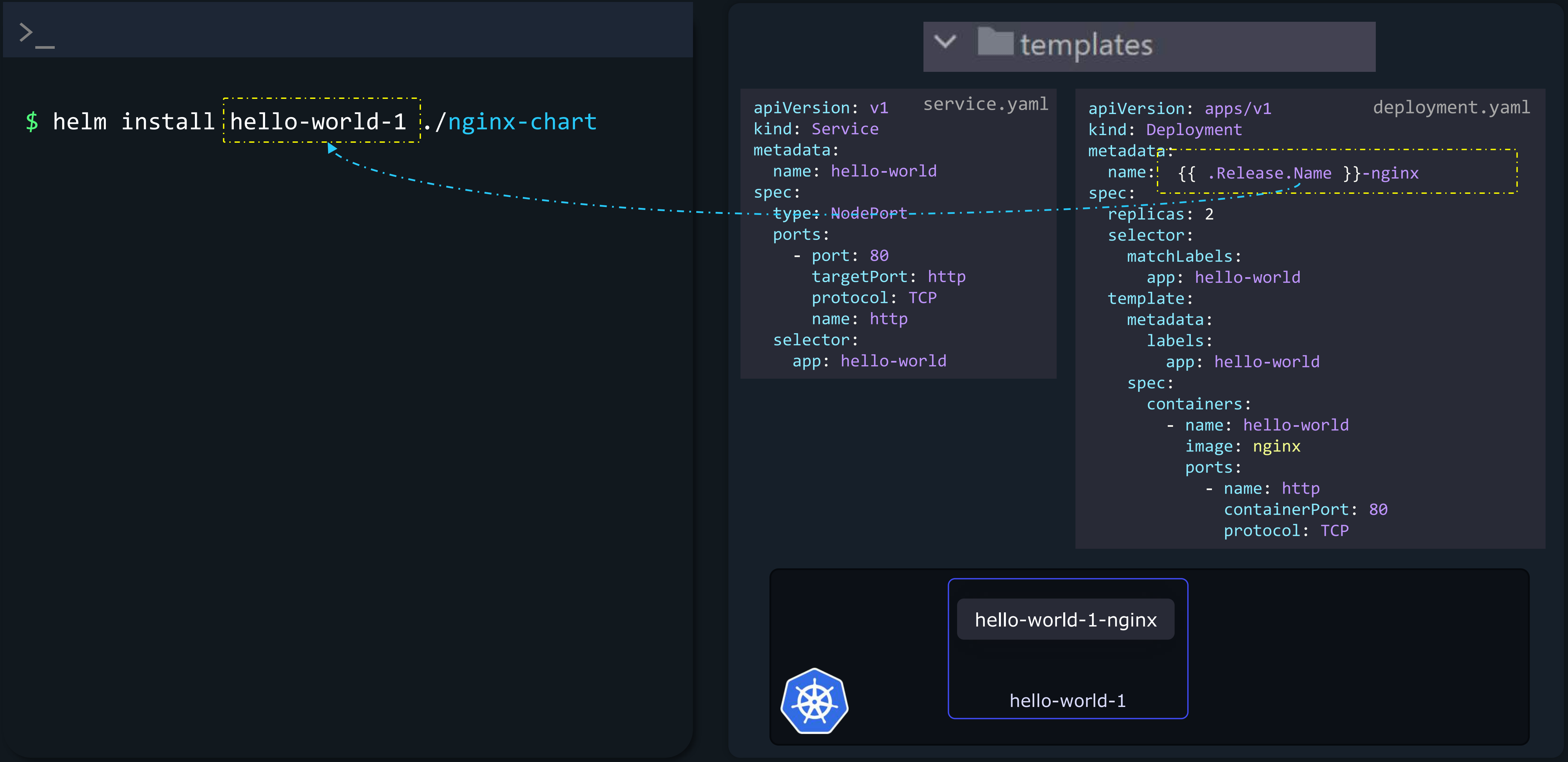
```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-world
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app: hello-world
```

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
```

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: {{ .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

Objects

Release.Name
Release.Namespace
Release.IsUpgrade
Release.IsInstall
Release.Revision
Release.Service

Chart.Name
Chart.ApiVersion
Chart.Version
Chart.Type
Chart.Keywords
Chart.Home

Capabilities.KubeVersion
Capabilities.ApiVersions
Capabilities.HelmVersion
Capabilities.GitCommit
Capabilities.GitTreeState
Capabilities.GoVersion

Values.replicaCount
Values.image

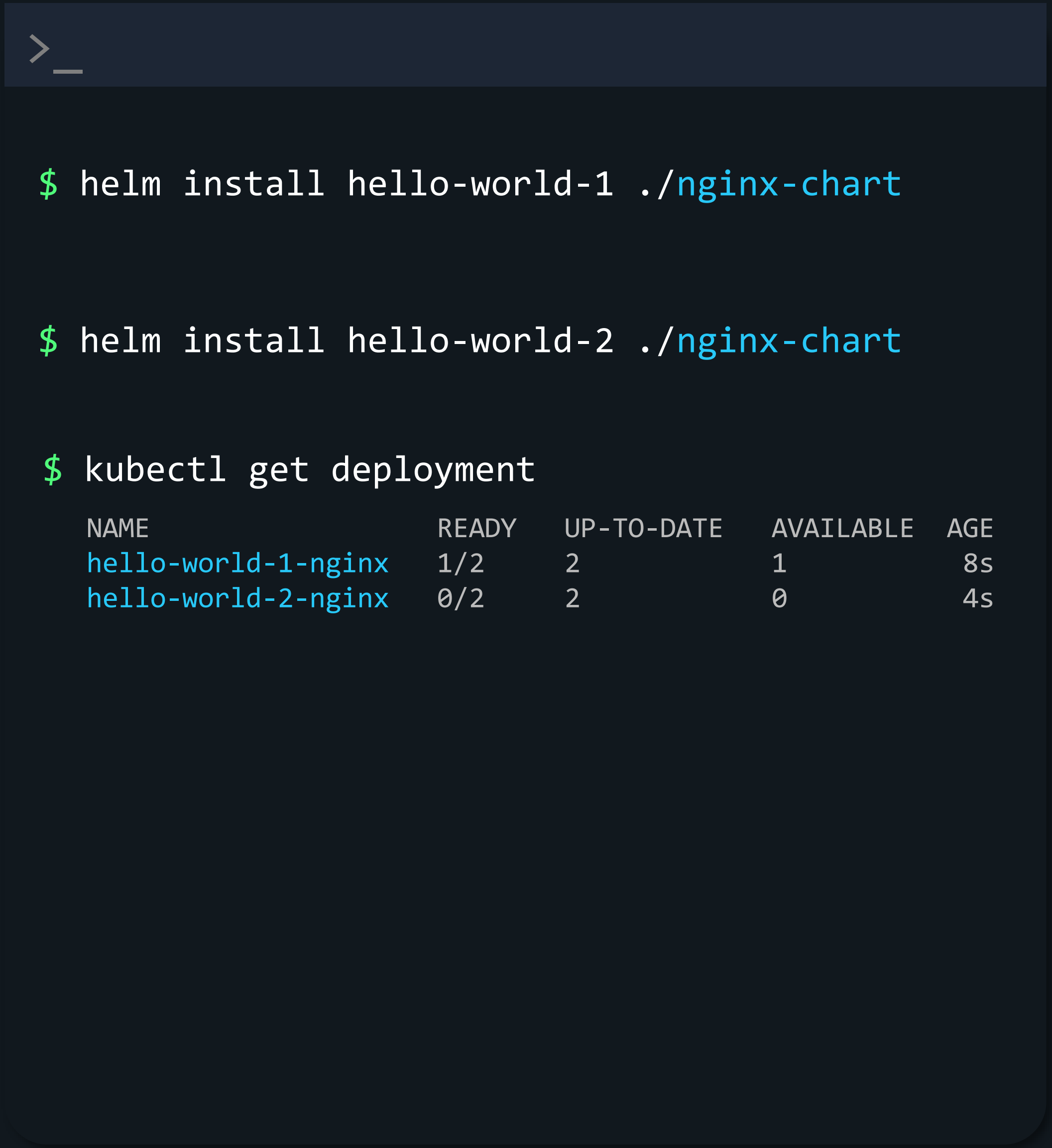
Chart.yaml

```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>
```

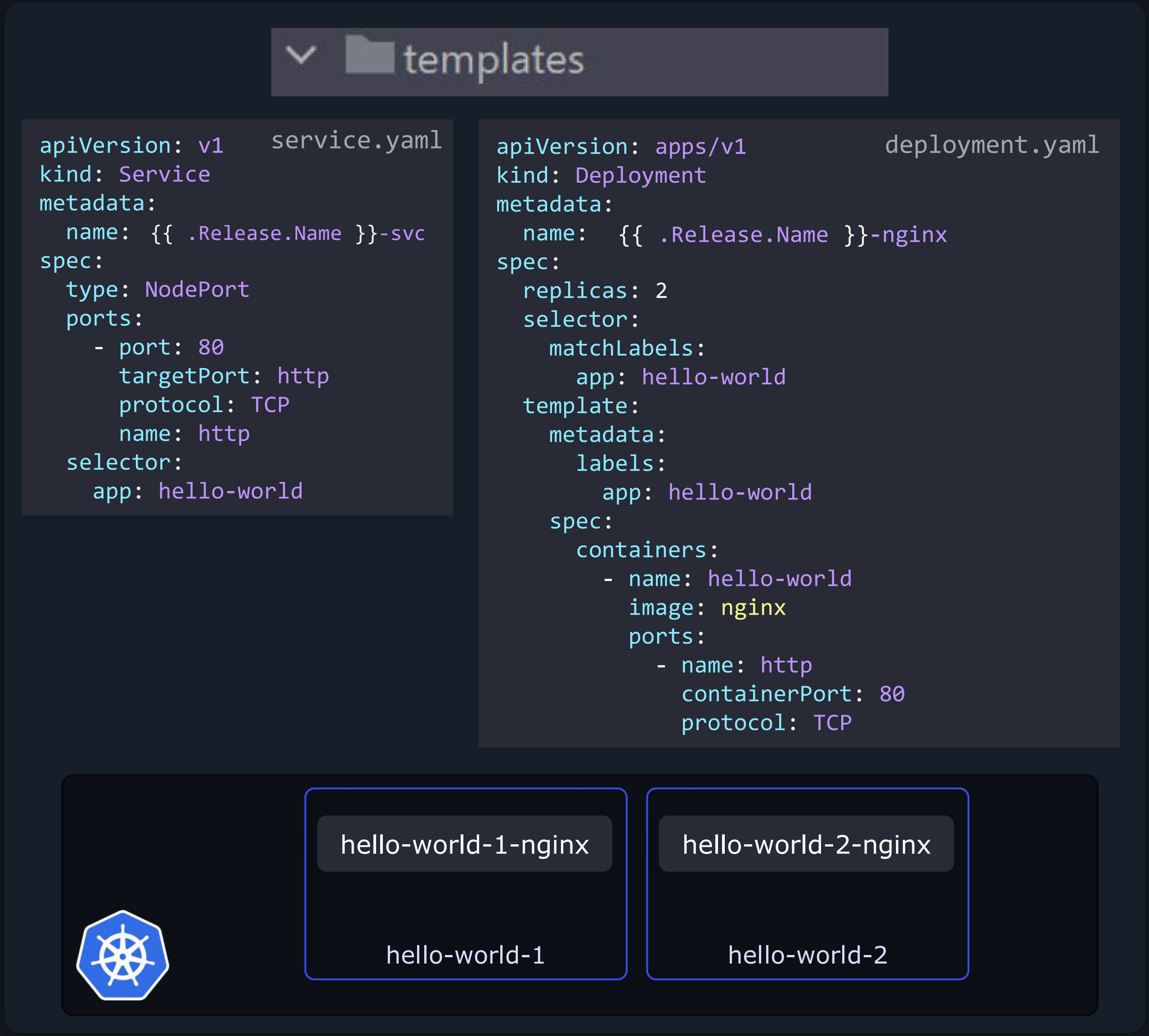
values.yaml

```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



```
$ 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
```



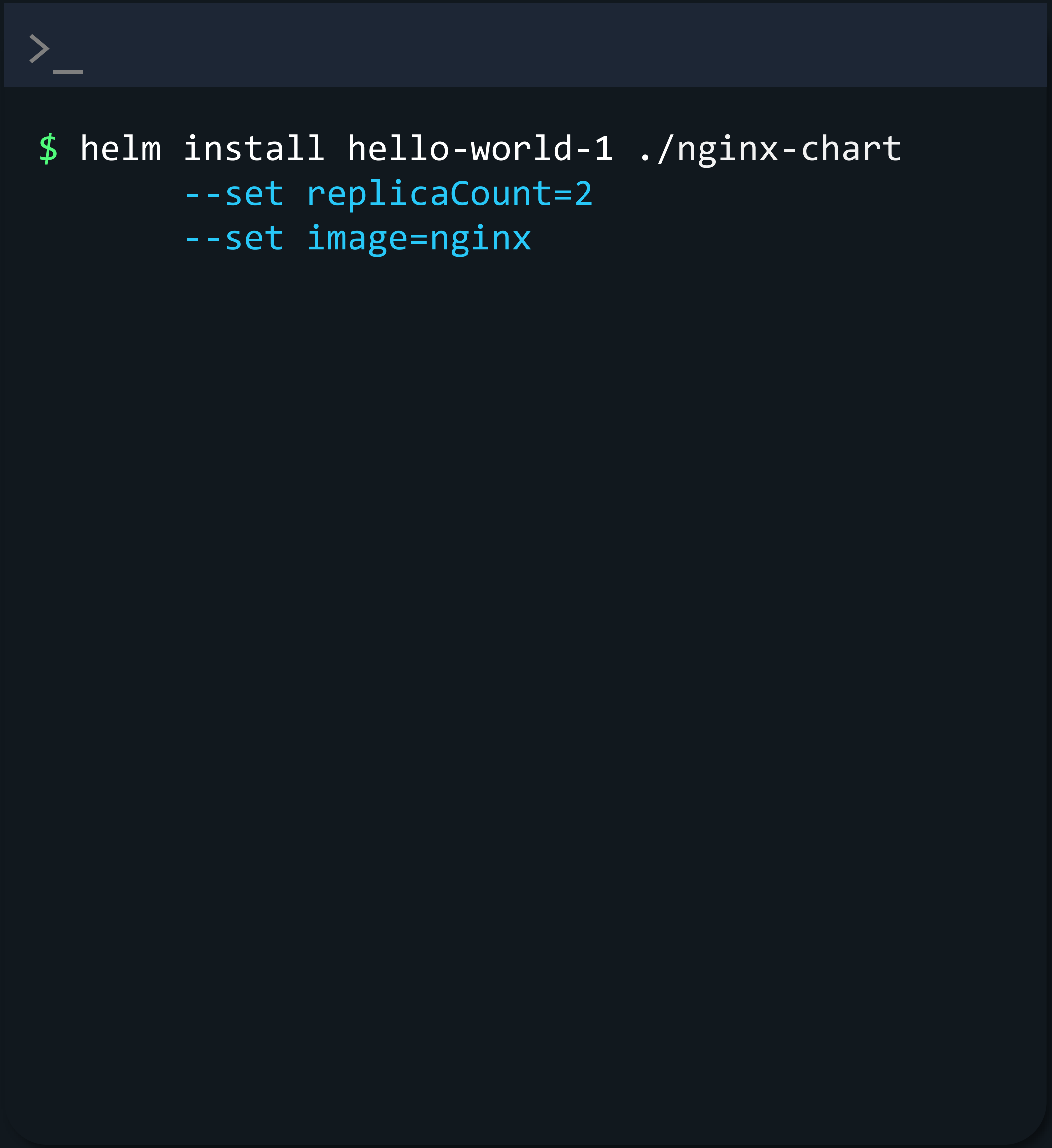
service.yaml

```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
```

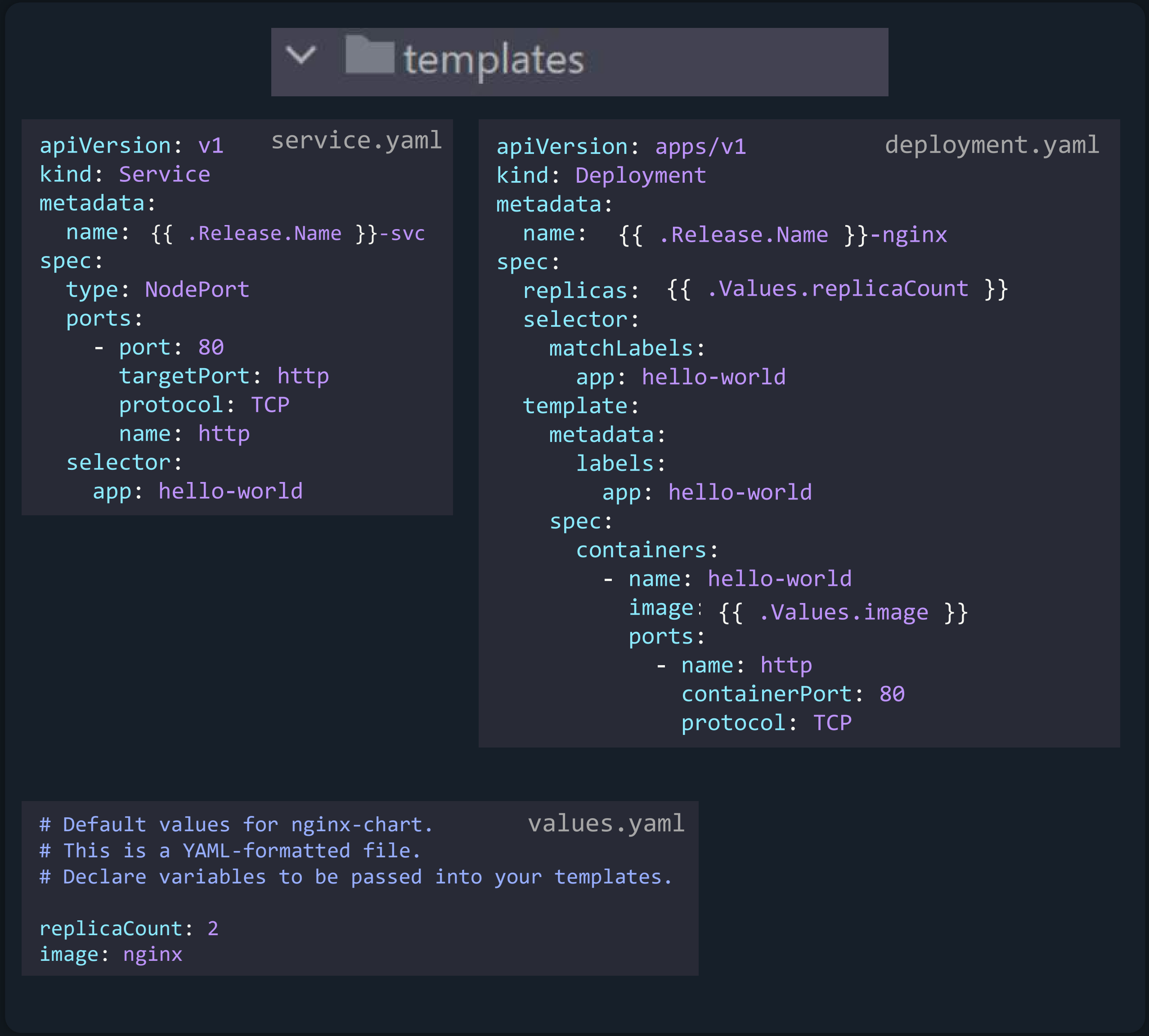
deployment.yaml

```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

templates

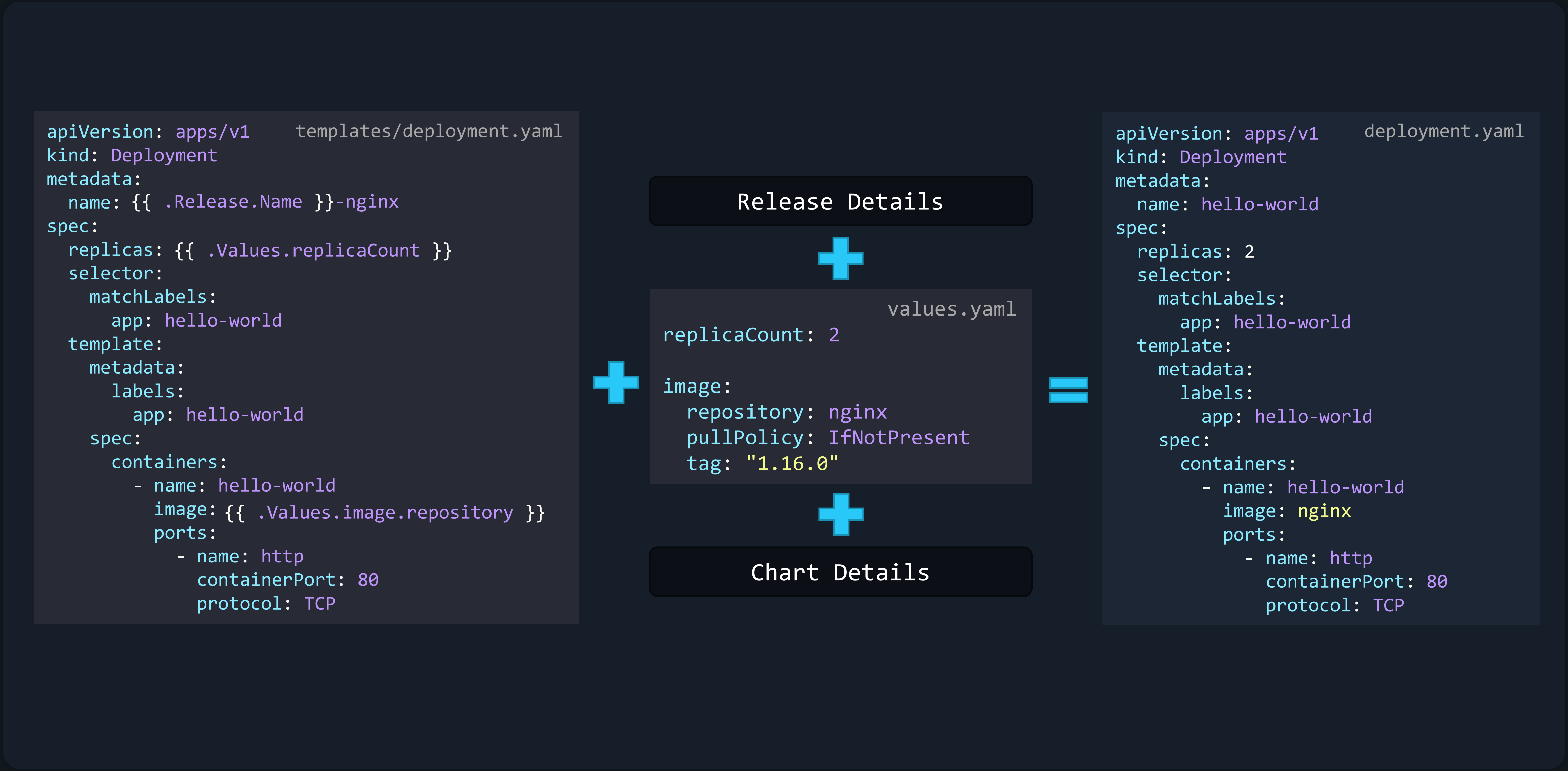
values.yaml

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

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

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 }}:{{ .Values.image.tag }}
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

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
```

values.yaml

```yaml
replicaCount: 2

image:
  repository: nginx
  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: nginx
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
```

Making Sure Chart is Working as Intended

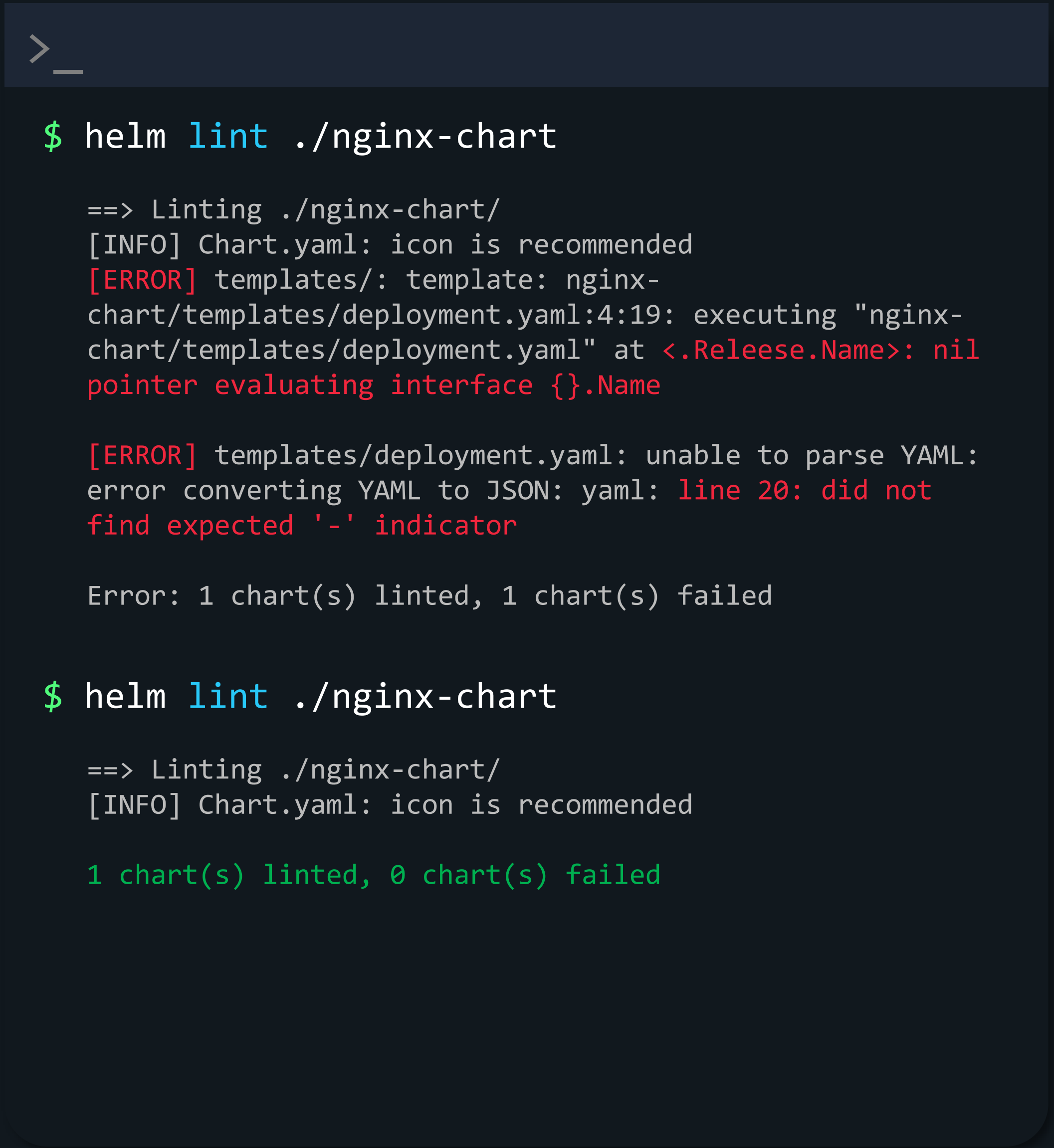
Verifying Helm Charts

Lint

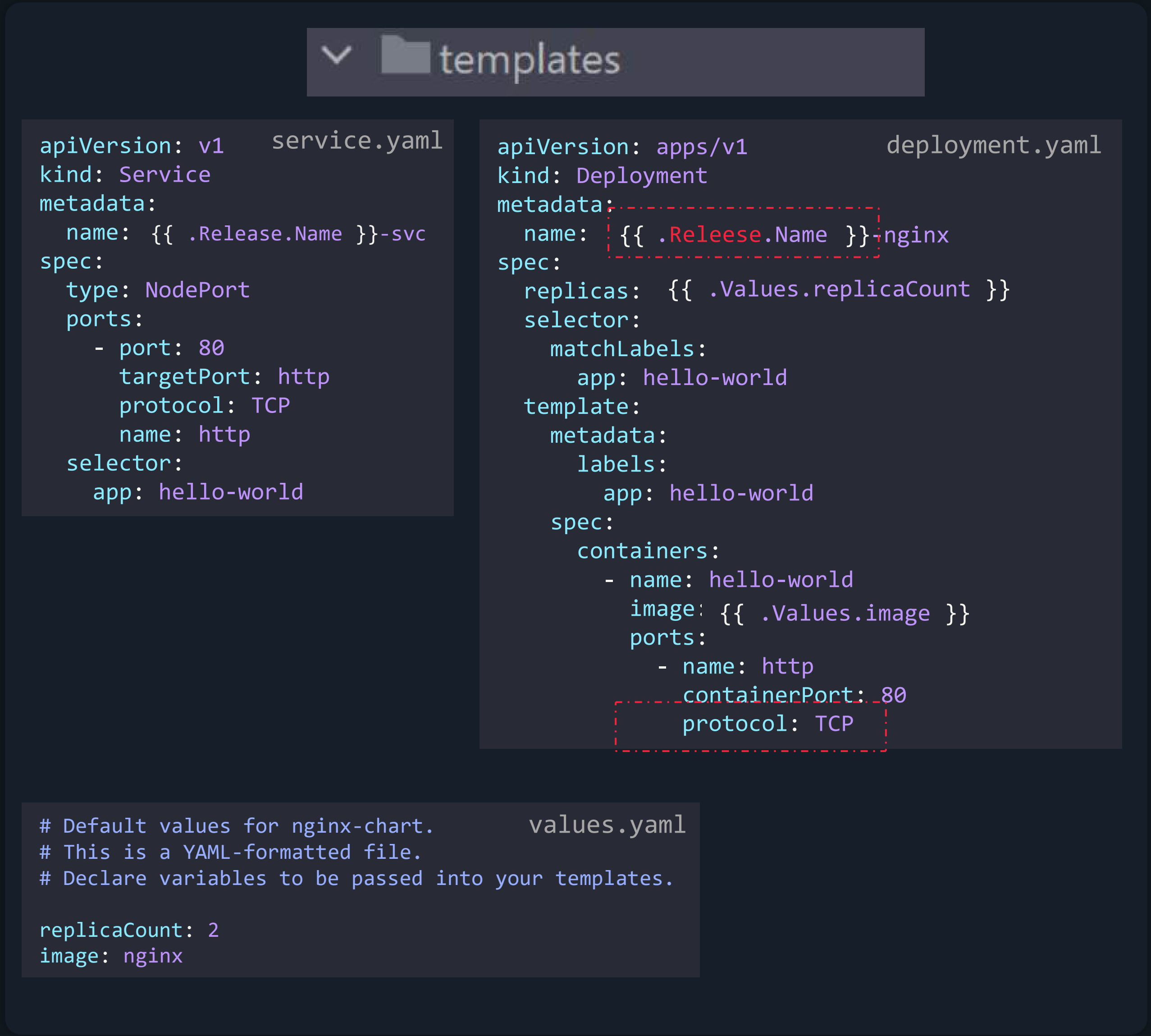
Template

Dry Run

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: {{ .Releese.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
```

KODEKLOUD

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
```

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
```

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

```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
```

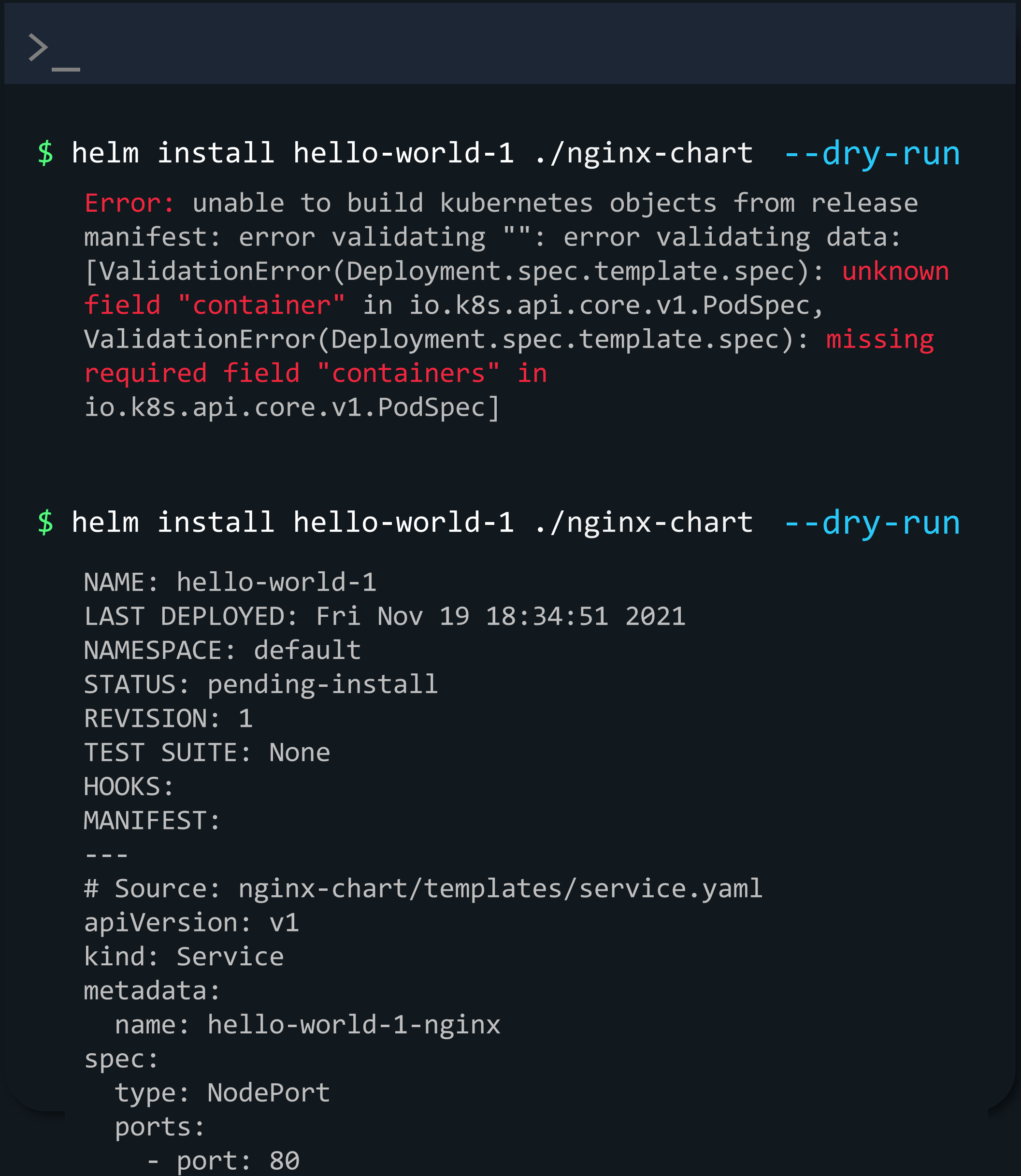
deployment.yaml

```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
```

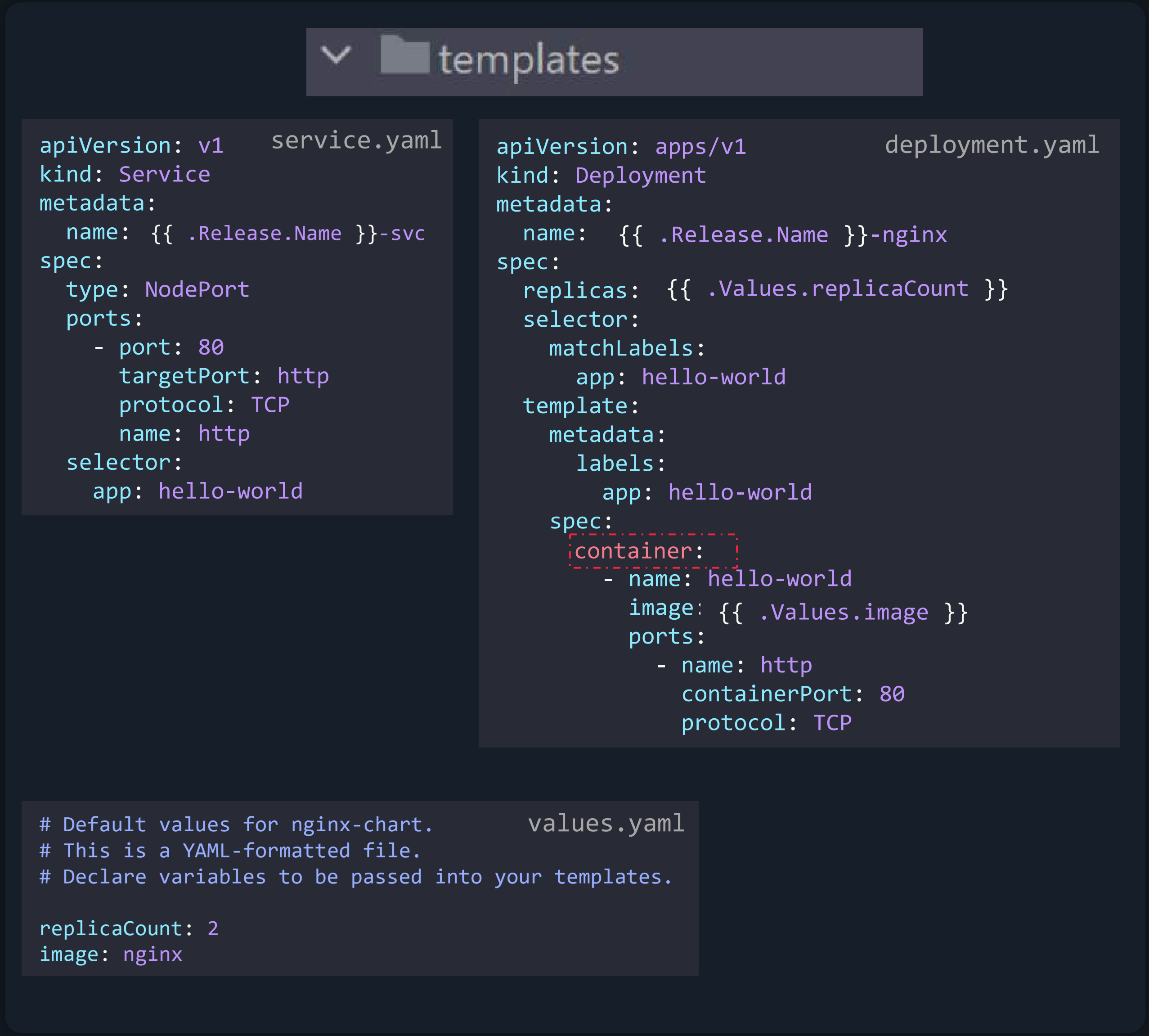
values.yaml

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

replicaCount: 2
image: nginx
```

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
```



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:
      container:
        - 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
```

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"
```

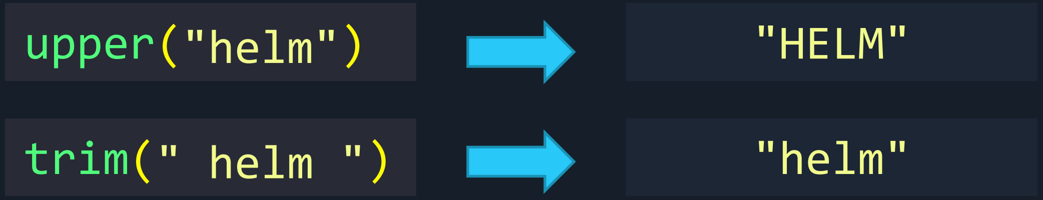
=

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
```

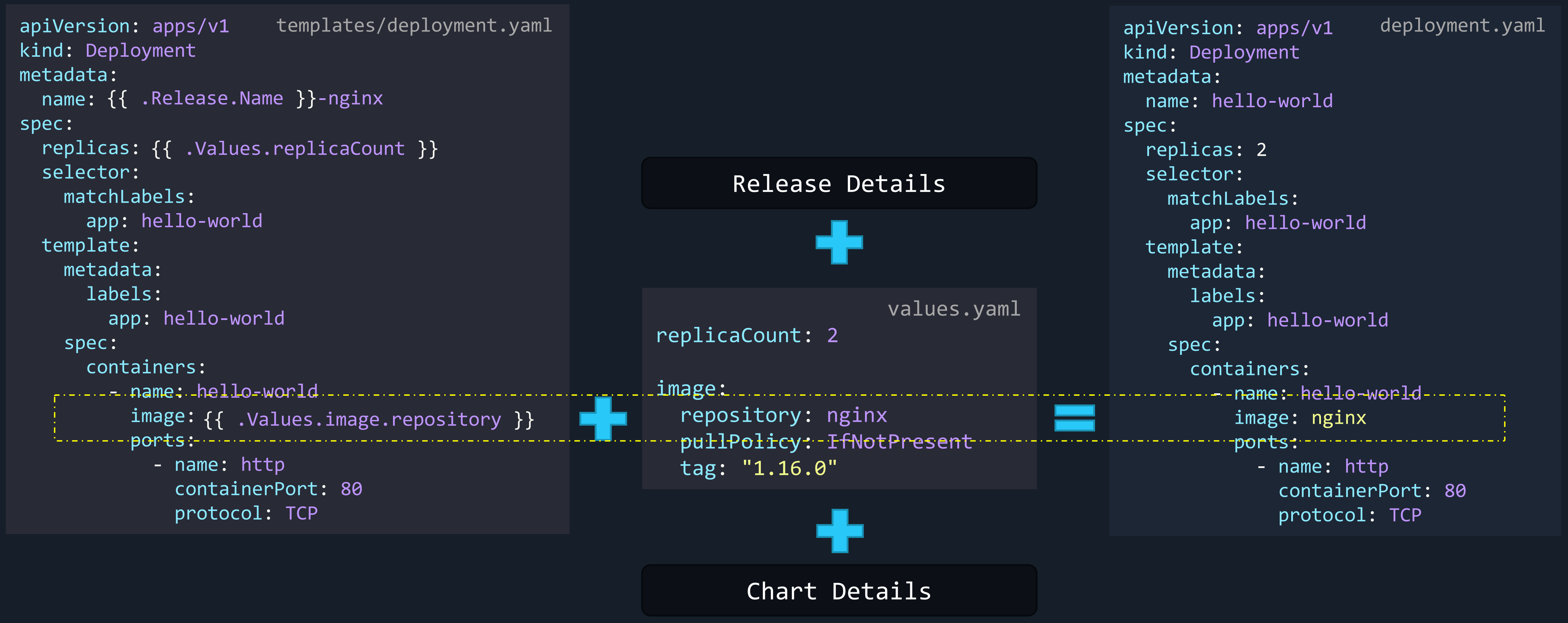
```
$ kubectl get pods
NAME                               READY   STATUS             RESTARTS   AGE
nginx-deployment-6c76ffbdd7-z4qgf  0/1     InvalidImageName   0          3s
```

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: "nginy"

{{  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

https://helm.sh/docs/chart_template_guide/function_list/

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

```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

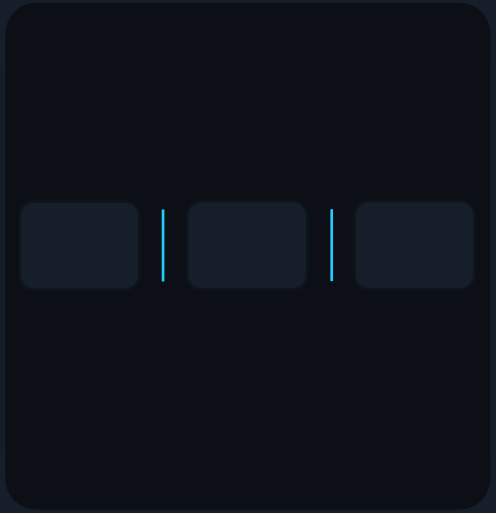
```yaml
replicaCount: 2

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

=

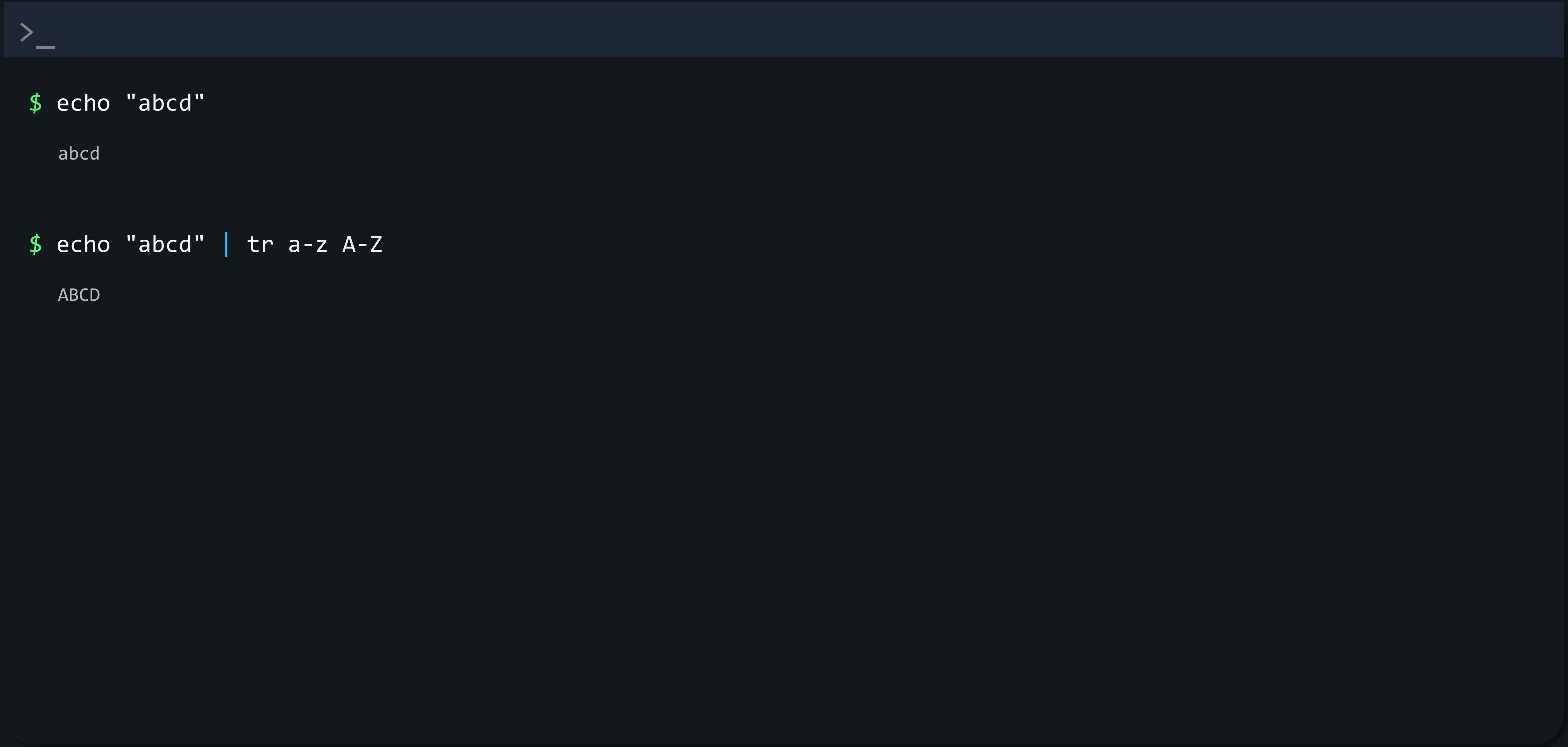
deployment.yaml

```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
```

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

```
replicaCount: 2
image: nginx

orgLabel: payroll
```

service.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 }}
  {{- end }}
spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```

service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: RELEASE-NAME-nginx

  labels:
    org: payroll

spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```

Conditionals

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
  labels:
     org: payroll
spec:
  ports:
    - port: 80
      name: http
  selector:
    app: hello-world
```

Conditionals

```
orgLabel = "payroll"

if orgLabel:
    print(orgLabel)
else if orgLabel=="hr":
    print("human resources")
else:
    print("nothing")
end
```

Conditionals

service.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

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
```

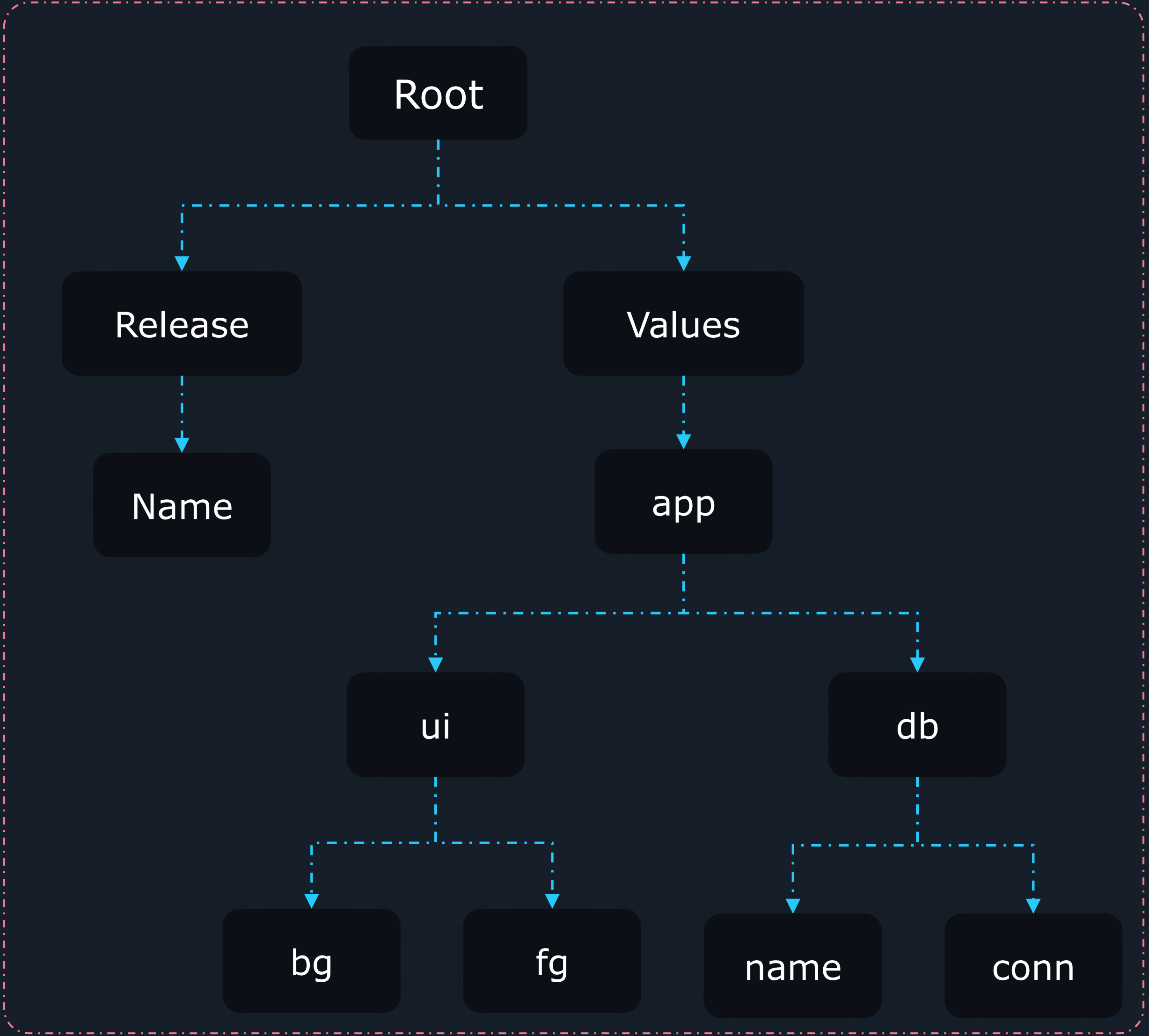
serviceaccount.yaml

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

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

{{- else }}
```

Setting scope using With

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

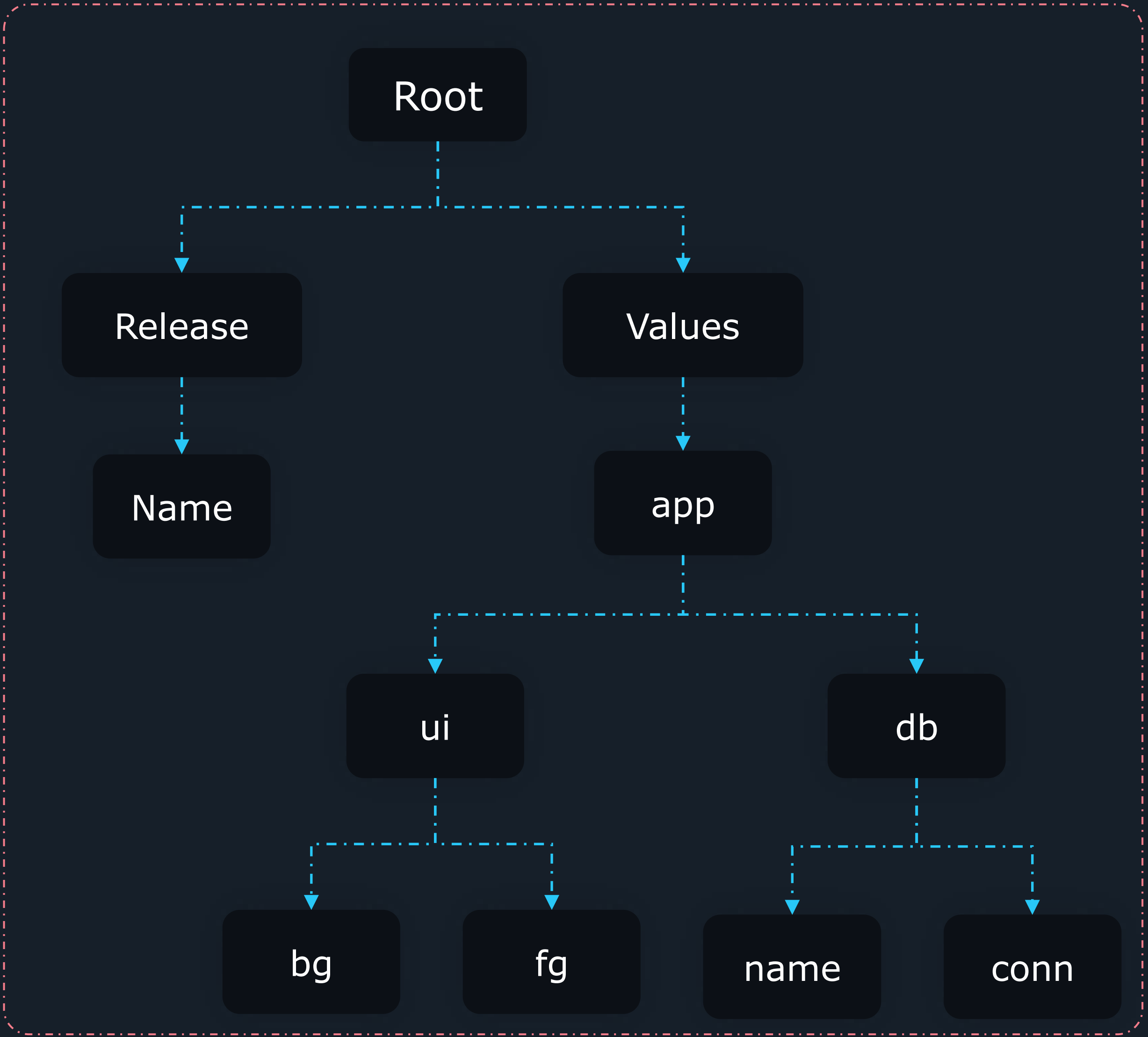
```yaml
# configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name:       {{ .Release.Name }}-appinfo
data:
  background: {{ .Values.app.ui.bg }}
  foreground: {{ .Values.app.ui.fg }}
  database:   {{ .Values.app.db.name }}
  connection: {{ .Values.app.db.conn }}
```



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 }}
```

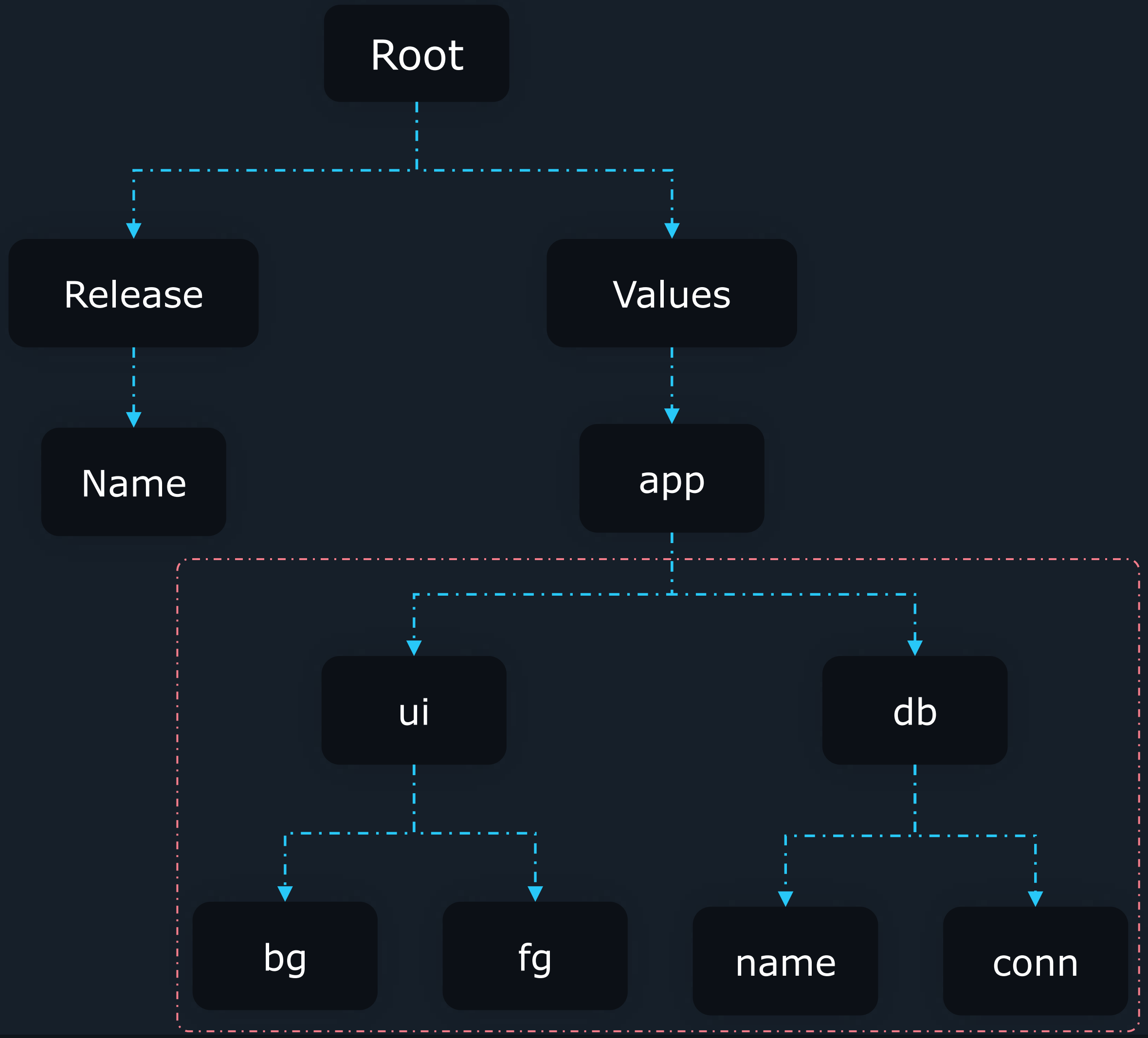
With



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 }}
```

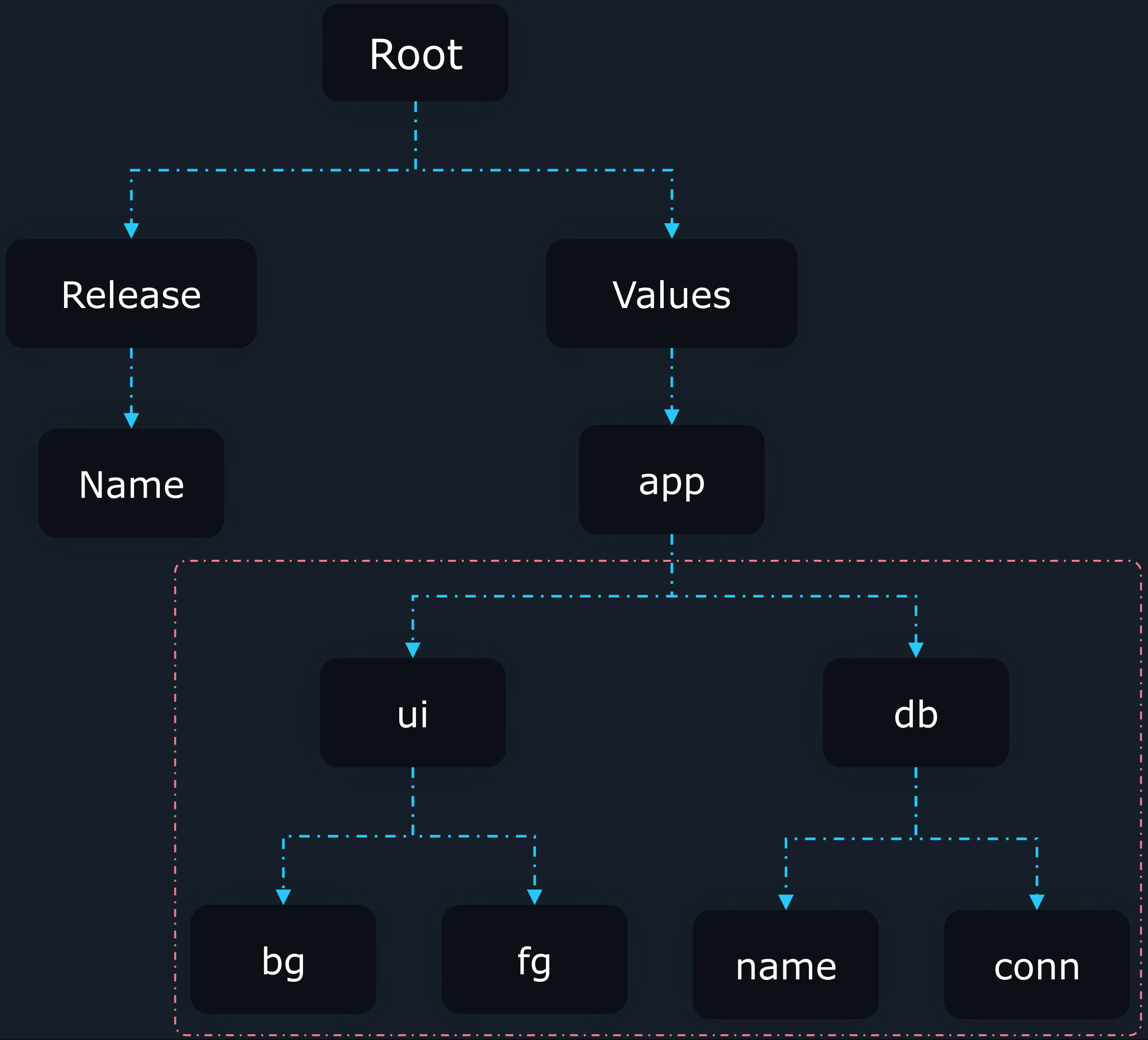
With



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 }}
```

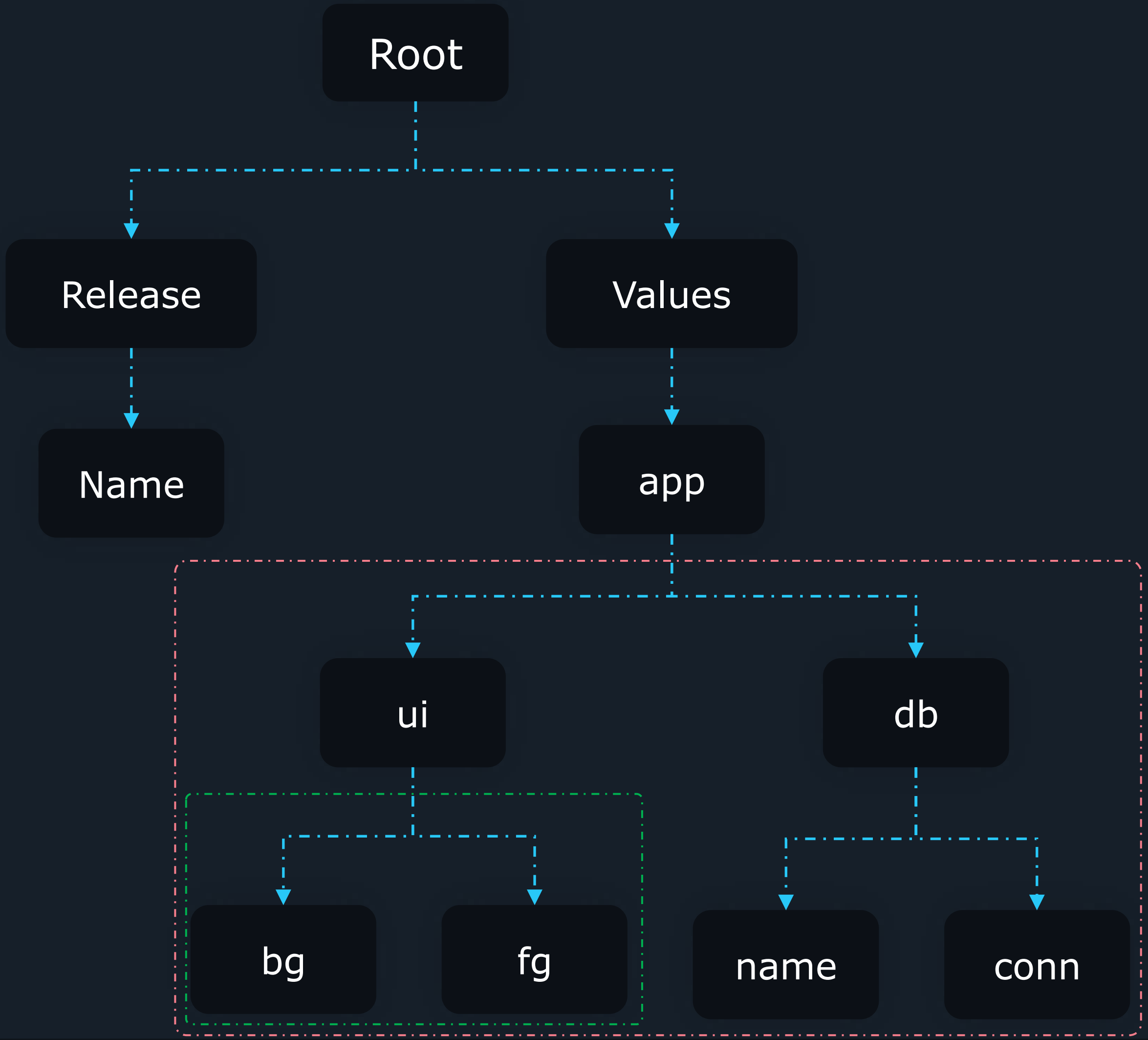
With



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 }}
```

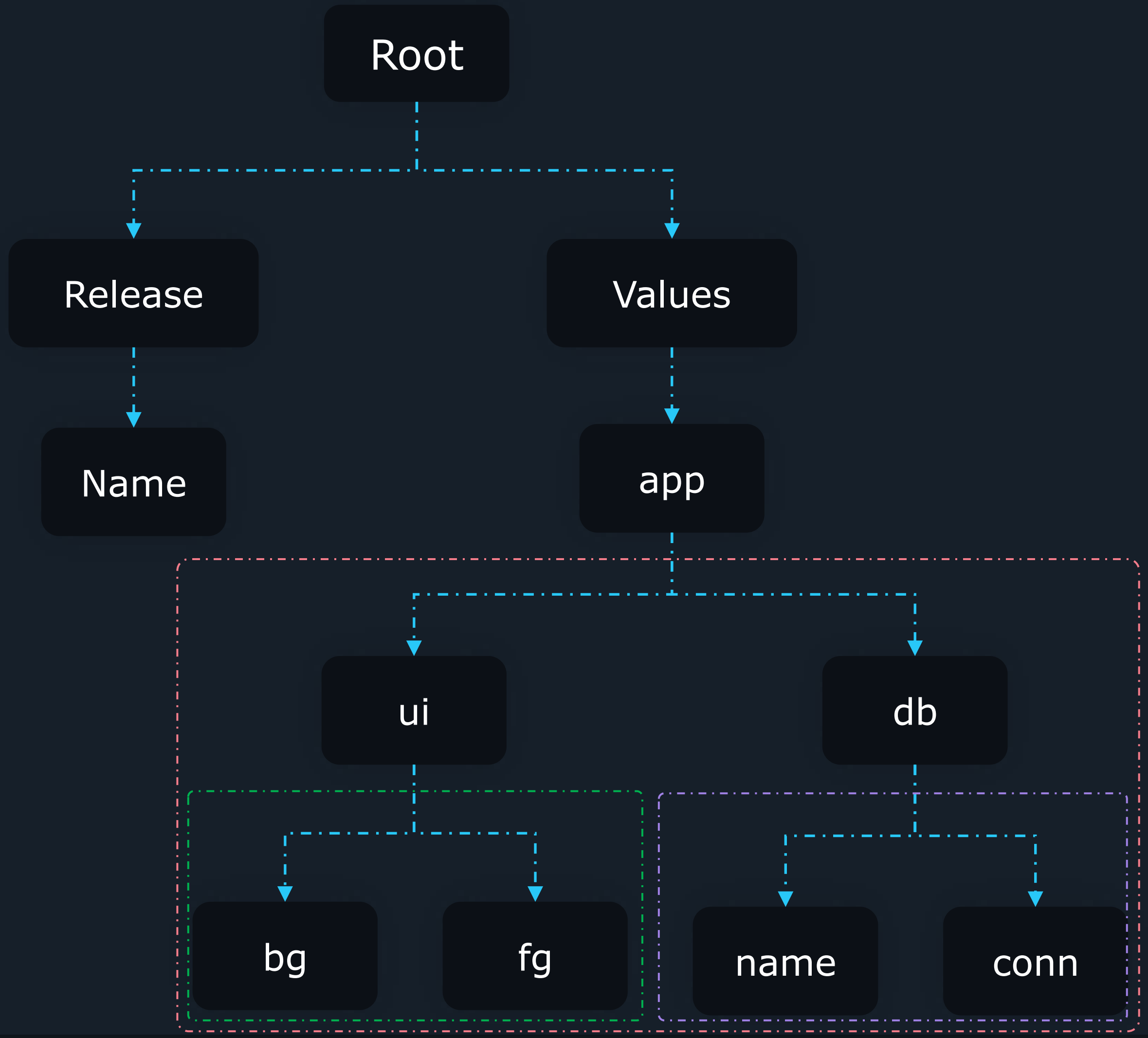
With



configmap.yaml

```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 }}
```

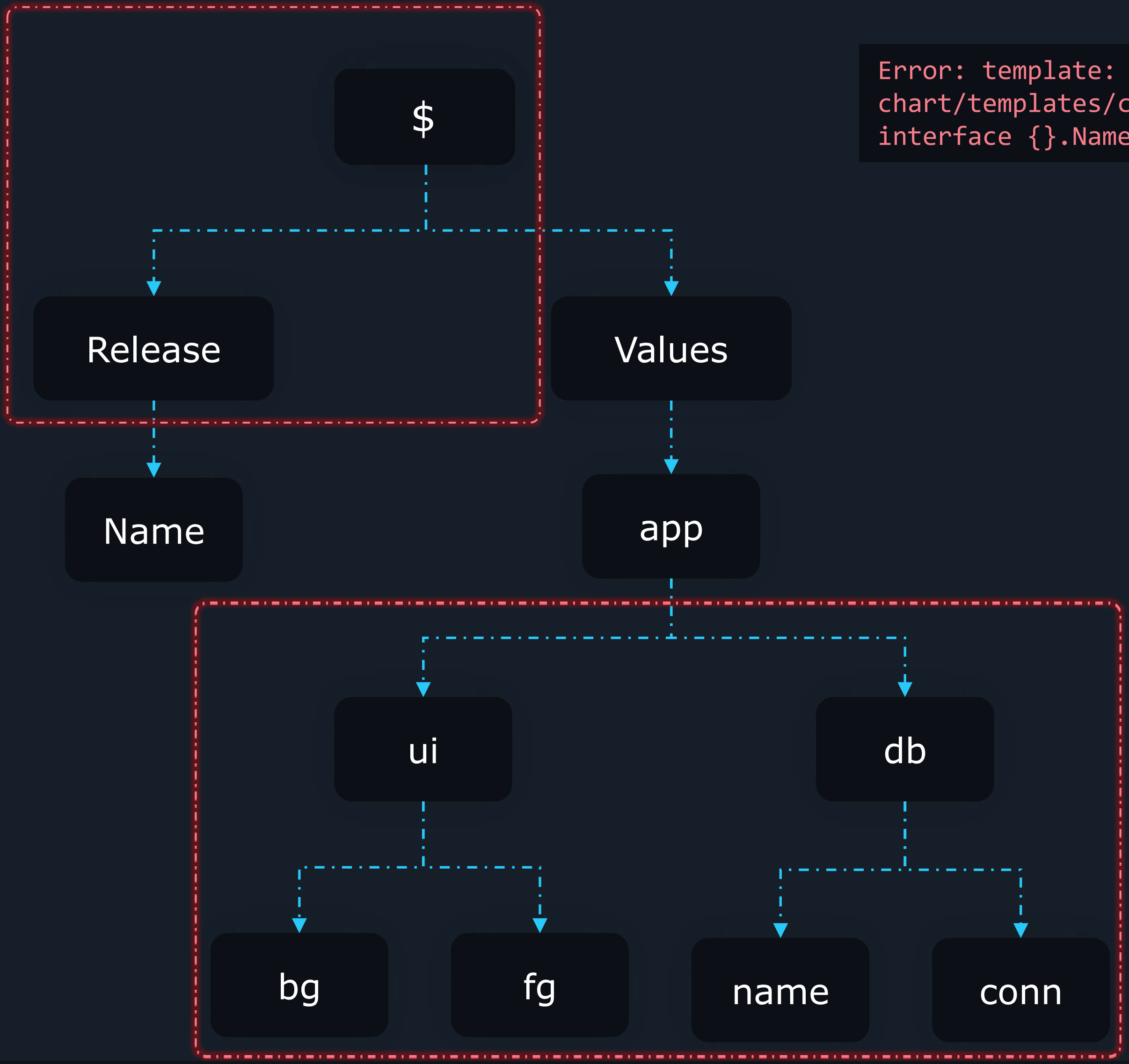
With



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 }}
```

With



```
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

values.yaml

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

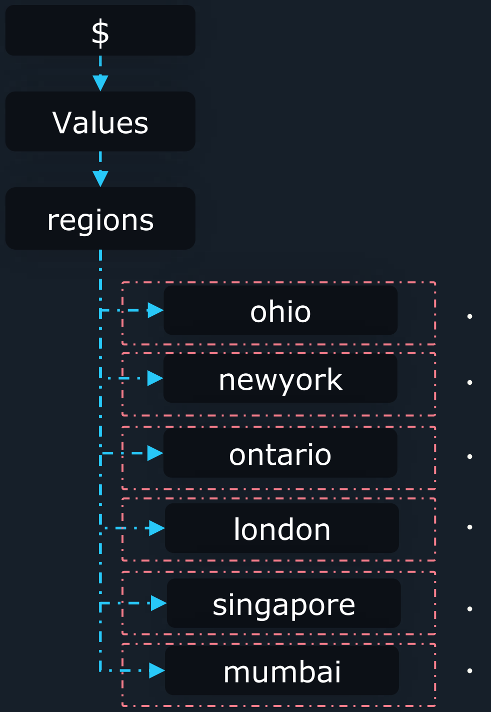
configmap.yaml

```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
```



$

Values

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

```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

```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

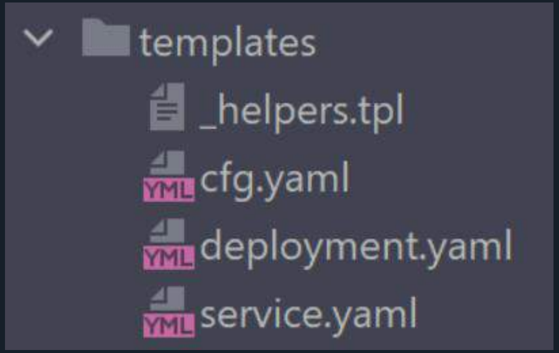
```
{{- define "labels" }}
app.kubernetes.io/name: {{ .Release.Name }}
app.kubernetes.io/instance: {{ .Release.Name }}
{{- end }}
```

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
```



```
templates
    _helpers.tpl
    cfg.yaml
    deployment.yaml
    service.yaml
```

Template

_helpers.tpl

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

template	action
include	function

deployment.yaml

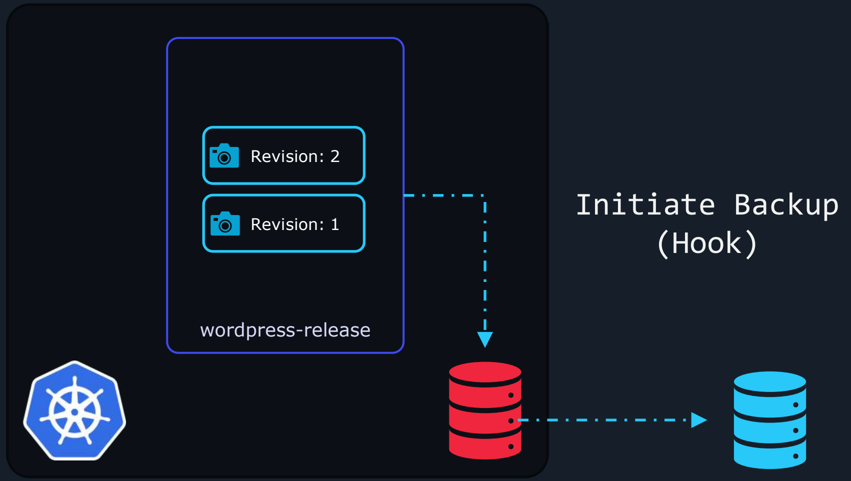
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}-nginx
  labels:
    {{- template "labels" .         }}

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
```

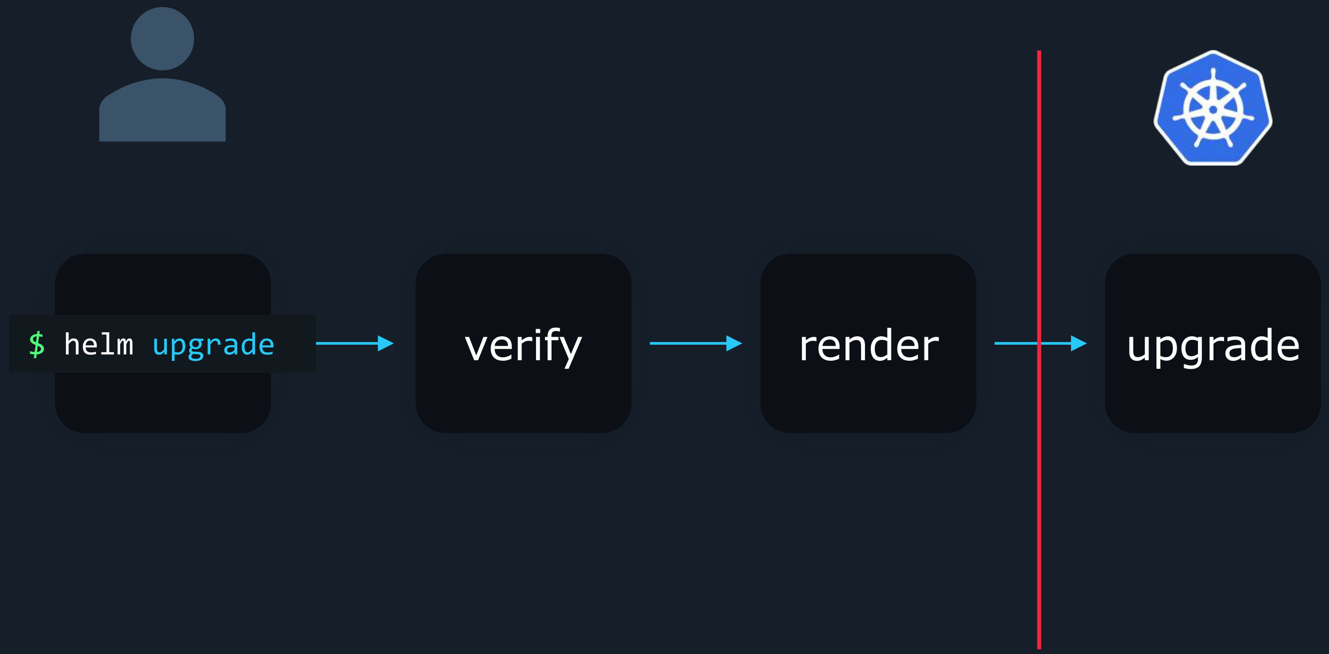
deployment.yaml

```
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
```

Chart Hooks

```
$ helm upgrade wordpress-release bitnami/wordpress
```





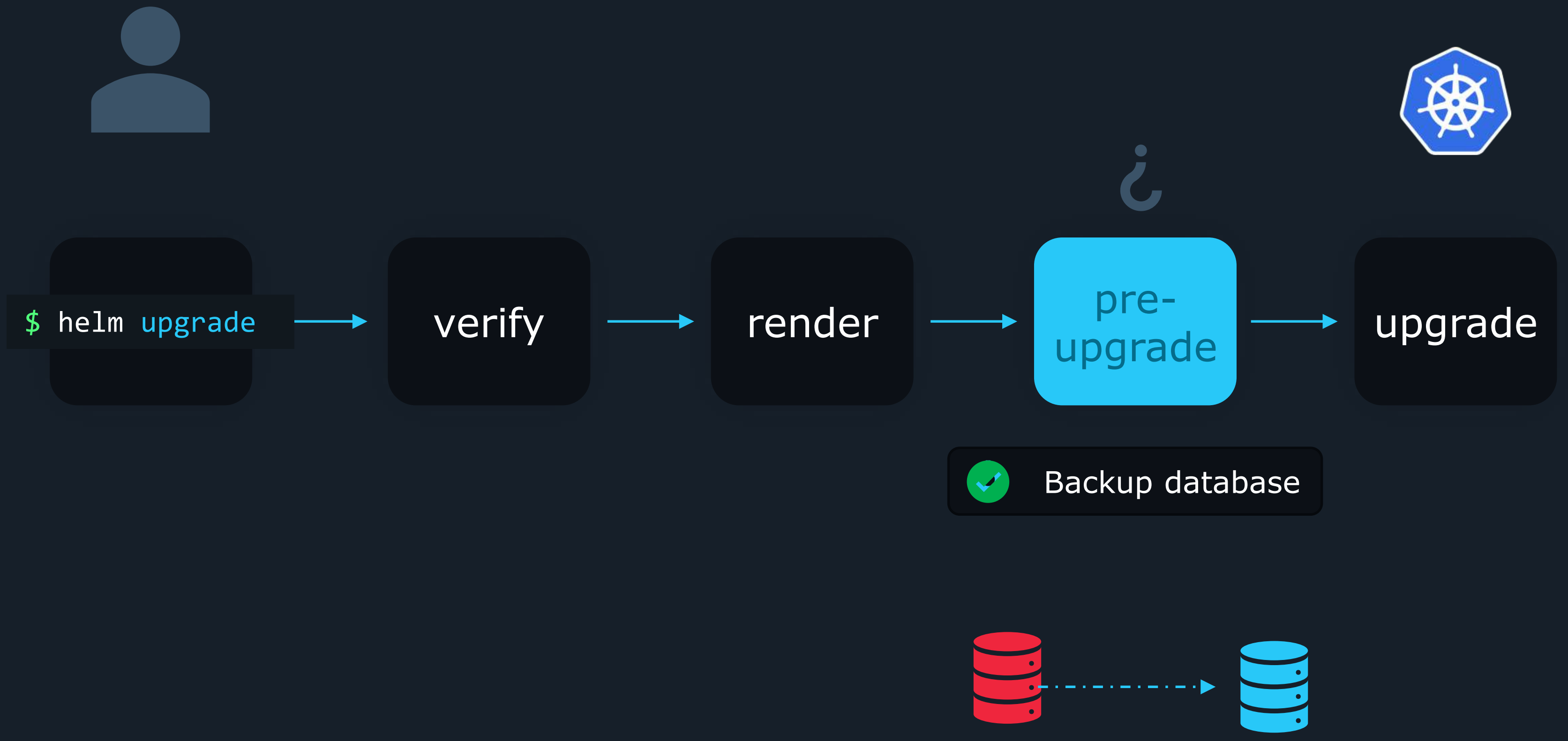
```
$ helm upgrade
```

verify

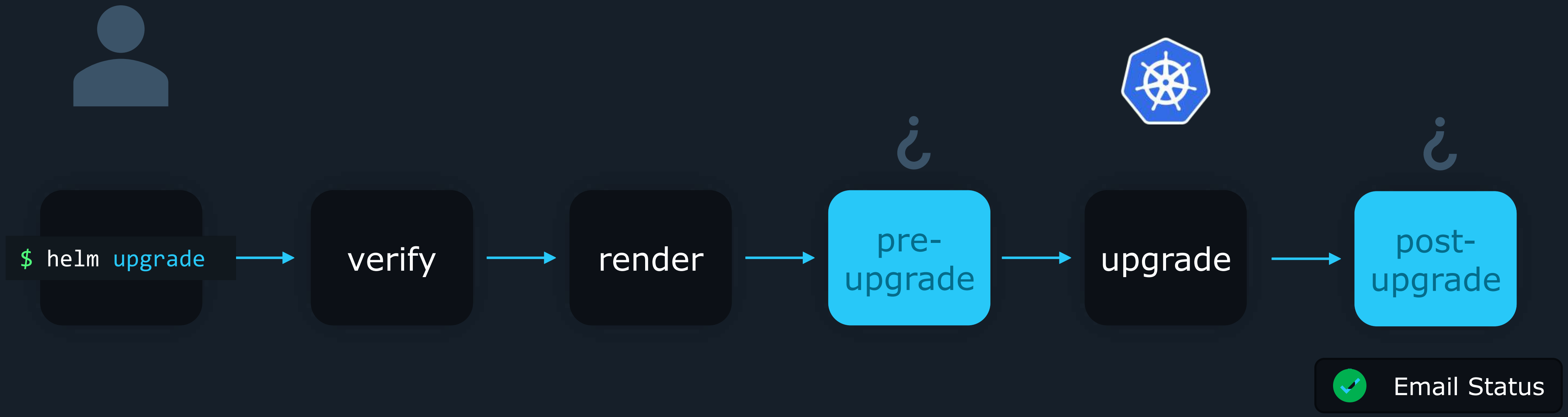
render

upgrade

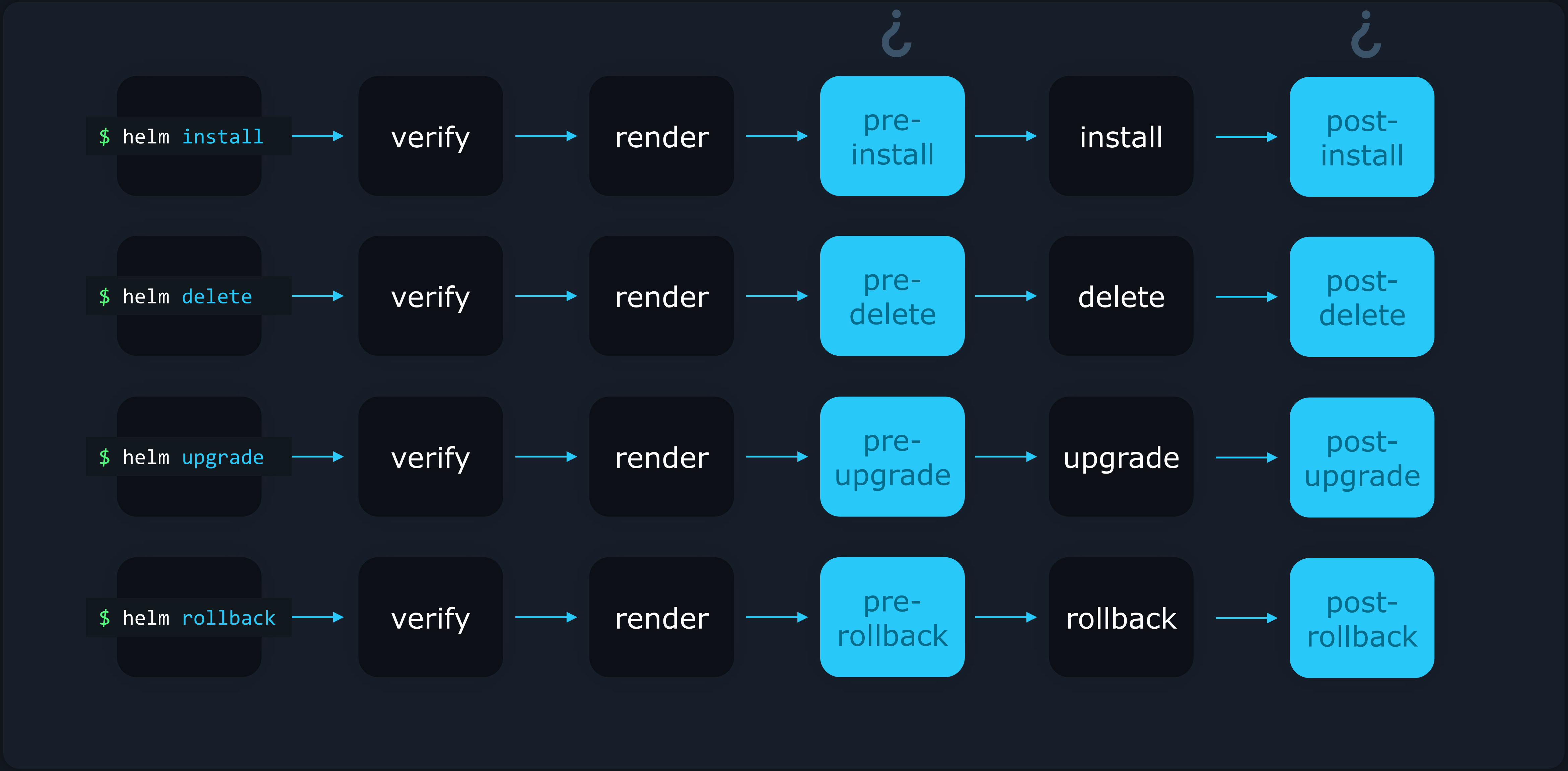
pre-upgrade hook



post-upgrade hook



hooks



```
$ helm install
```

verify → render → pre-install → install → post-install

```
$ helm delete
```

verify → render → pre-delete → delete → post-delete

```
$ helm upgrade
```

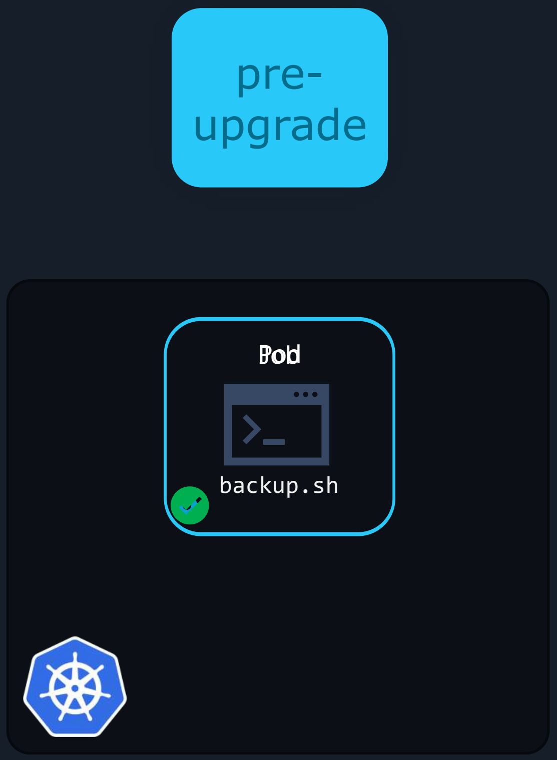
verify → render → pre-upgrade → upgrade → post-upgrade

```
$ helm rollback
```

verify → render → pre-rollback → rollback → post-rollback

Creating Hooks

- hello-world-chart
  - templates
    - service.yaml
    - deployment.yaml
    - secret.yaml
    - backup-job.yaml

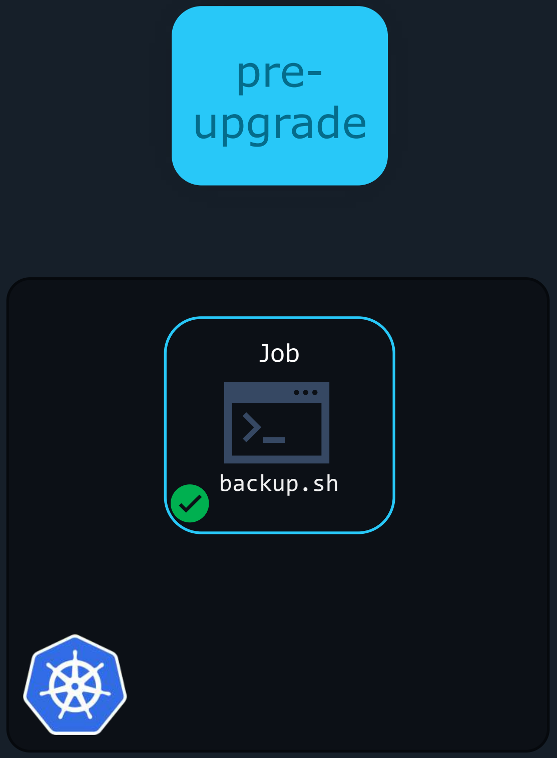


backup-job.yaml

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: {{ .Release.Name }}-nginx
spec:
  template:
    metadata:
      name: {{ .Release.Name }}-nginx
    spec:
      restartPolicy: Never
      containers:
      - name: pre-upgrade-backup-job
        image: "alpine"
        command: ["/bin/backup.sh"]
```

Creating Hooks

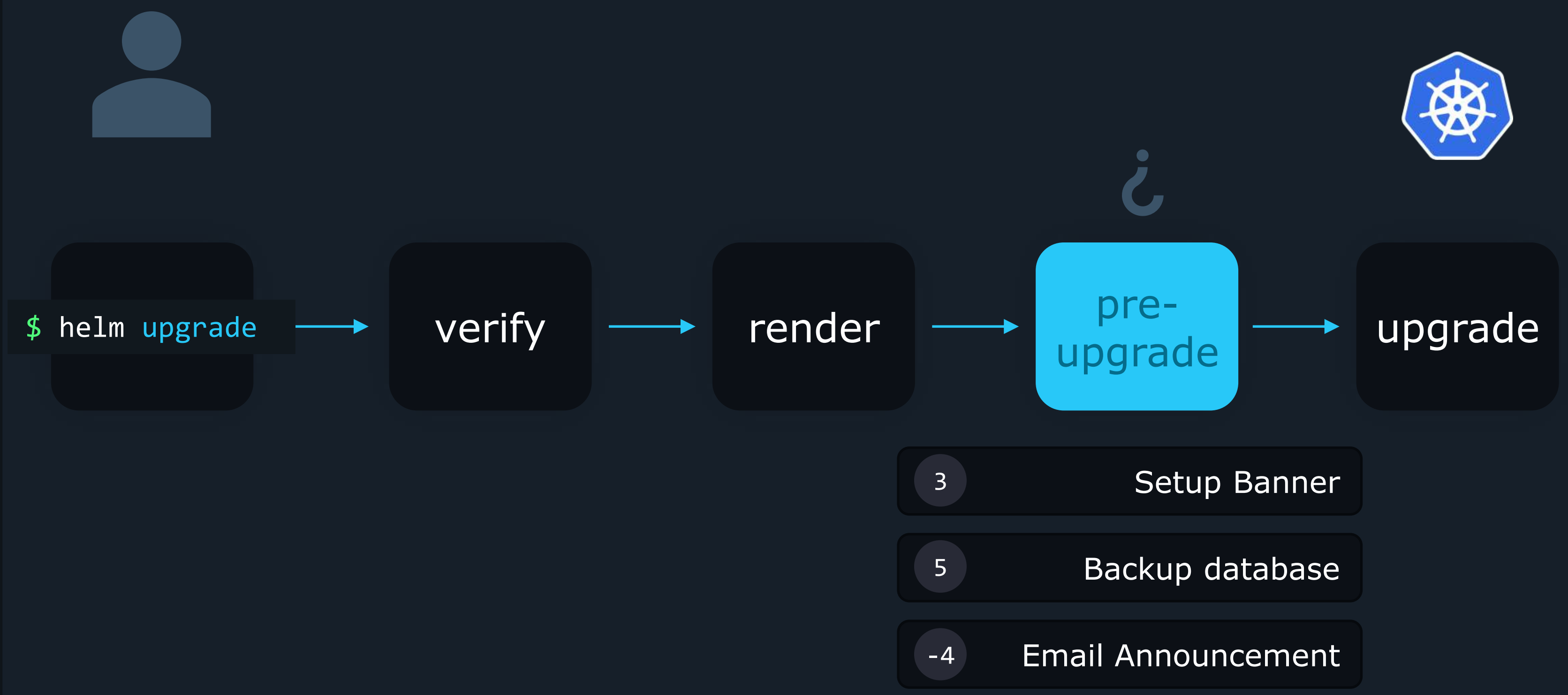
- hello-world-chart
  - templates
    - service.yaml
    - deployment.yaml
    - secret.yaml
    - backup-job.yaml



backup-job.yaml

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: {{ .Release.Name }}-nginx

  annotations:
    "helm.sh/hook": pre-upgrade


spec:
  template:
    metadata:
      name: {{ .Release.Name }}-nginx
    spec:
      restartPolicy: Never
      containers:
      - name: pre-upgrade-backup-job
        image: "alpine"
        command: ["/bin/backup.sh"]
```

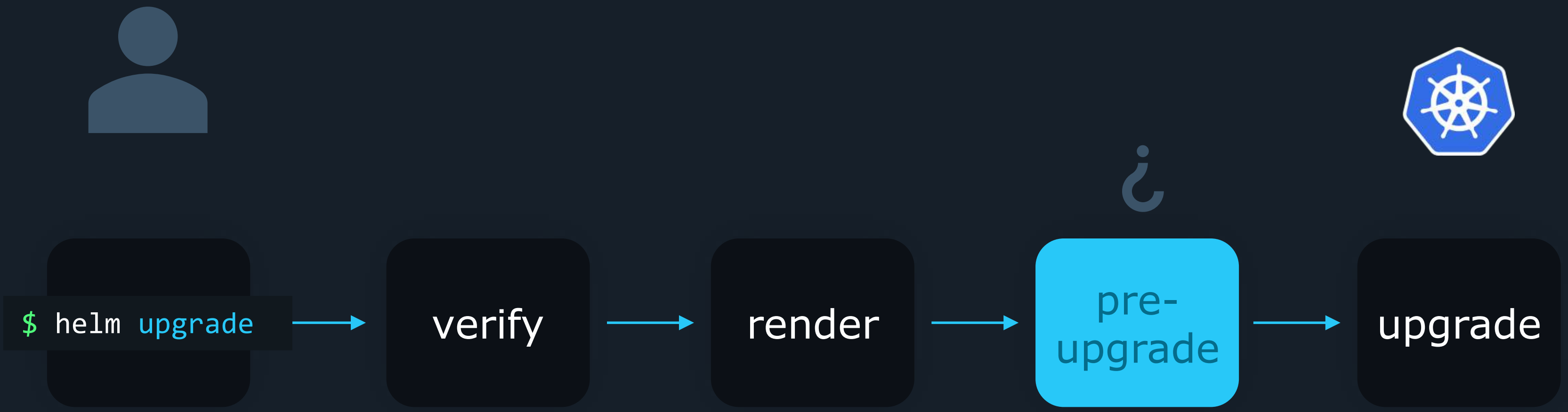
pre-upgrade hook



backup-job.yaml

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: {{ .Release.Name }}-nginx

  annotations:
    "helm.sh/hook": pre-upgrade



spec:
  template:
    metadata:
      name: {{ .Release.Name }}-nginx
    spec:
      restartPolicy: Never
      containers:
      - name: pre-upgrade-backup-job
        image: "alpine"
        command: ["/bin/backup.sh"]
```

pre-upgrade hook



-4	Email Announcement
3	Setup Banner
5	Backup database

backup-job.yaml

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: {{ .Release.Name }}-nginx

  annotations:
    "helm.sh/hook": pre-upgrade

    "helm.sh/hook-weight": "5"

    "helm.sh/hook-delete-policy": hook-succeeded
spec:
  template:
    metadata:
      name: {{ .Release.Name }}-nginx
    spec:
      restartPolicy: Never
      containers:
      - name: pre-upgrade-backup-job
        image: "alpine"
        command: ["/bin/backup.sh"]
```

hook-succeeded

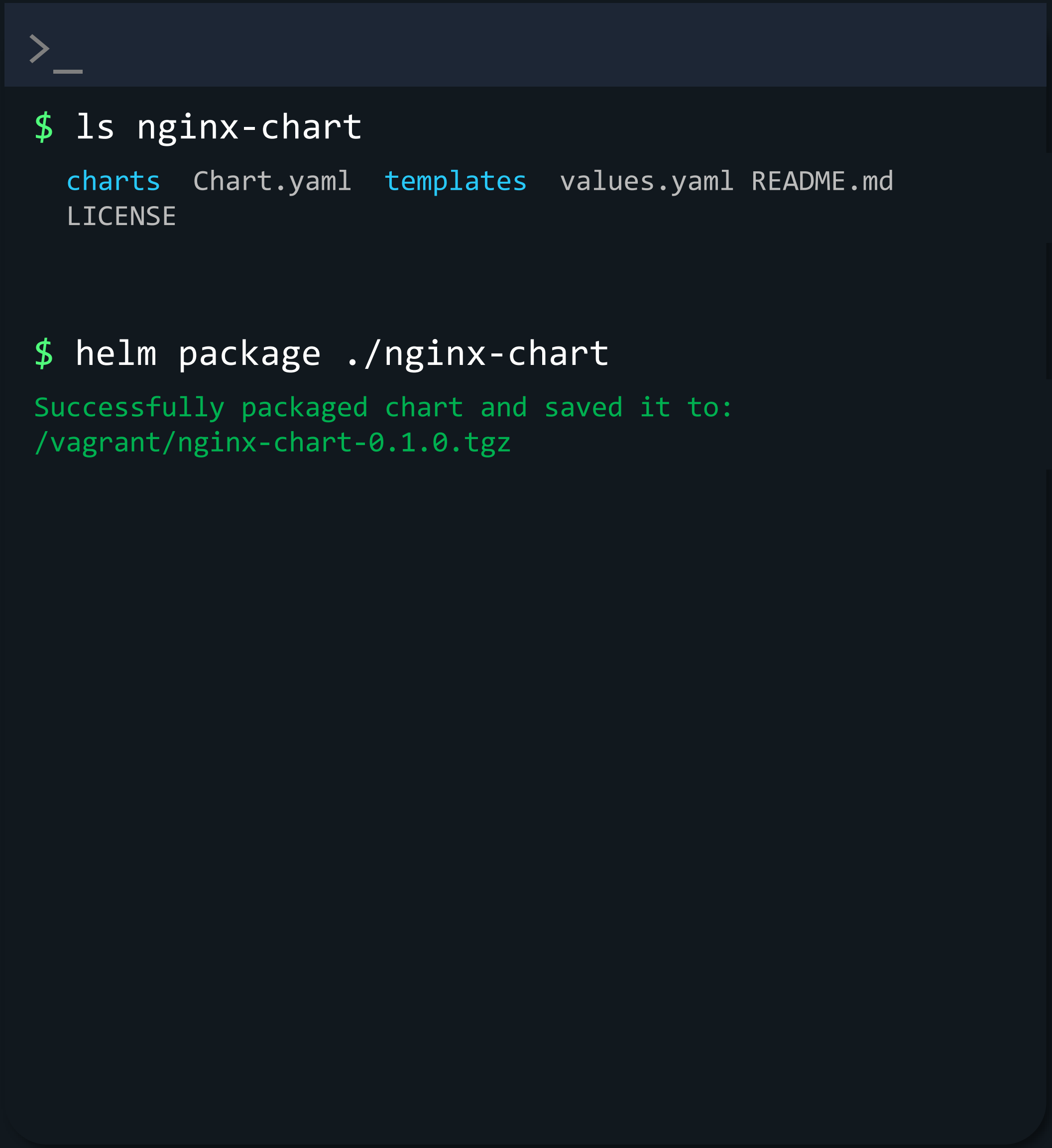
hook-failed

before-hook-creation

Packaging and Signing Charts

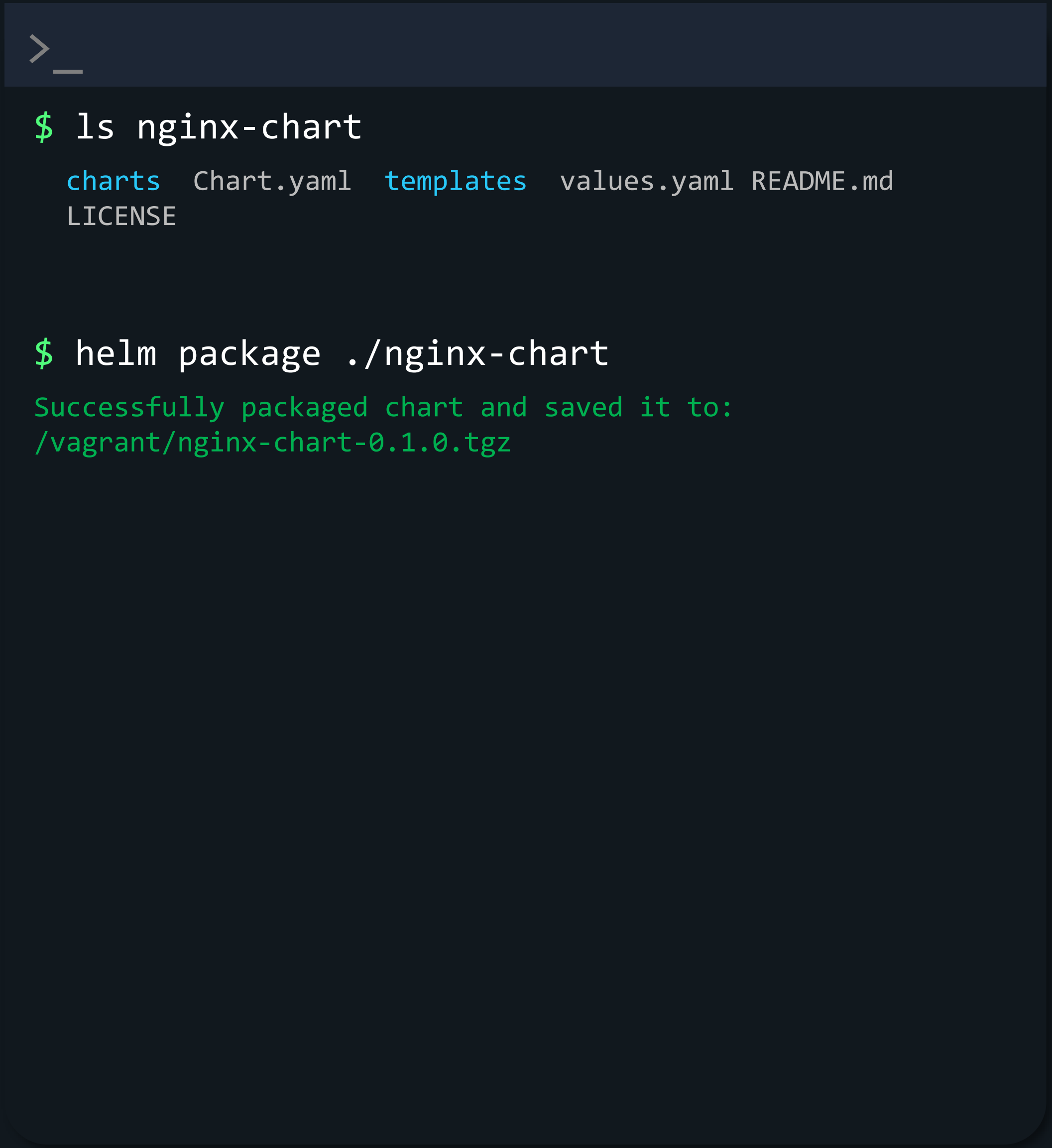
Packaging & Uploading

Online Chart Repository

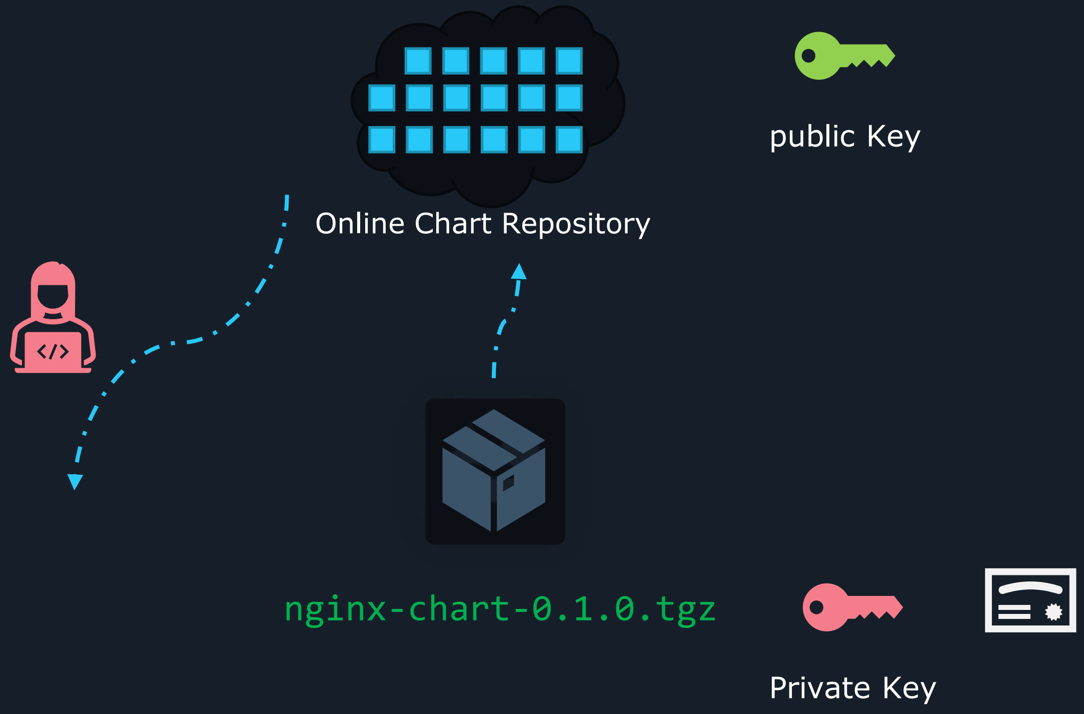


```
$ ls nginx-chart
  charts  Chart.yaml  templates  values.yaml README.md
  LICENSE


$ helm package ./nginx-chart
Successfully packaged chart and saved it to:
/vagrant/nginx-chart-0.1.0.tgz
```

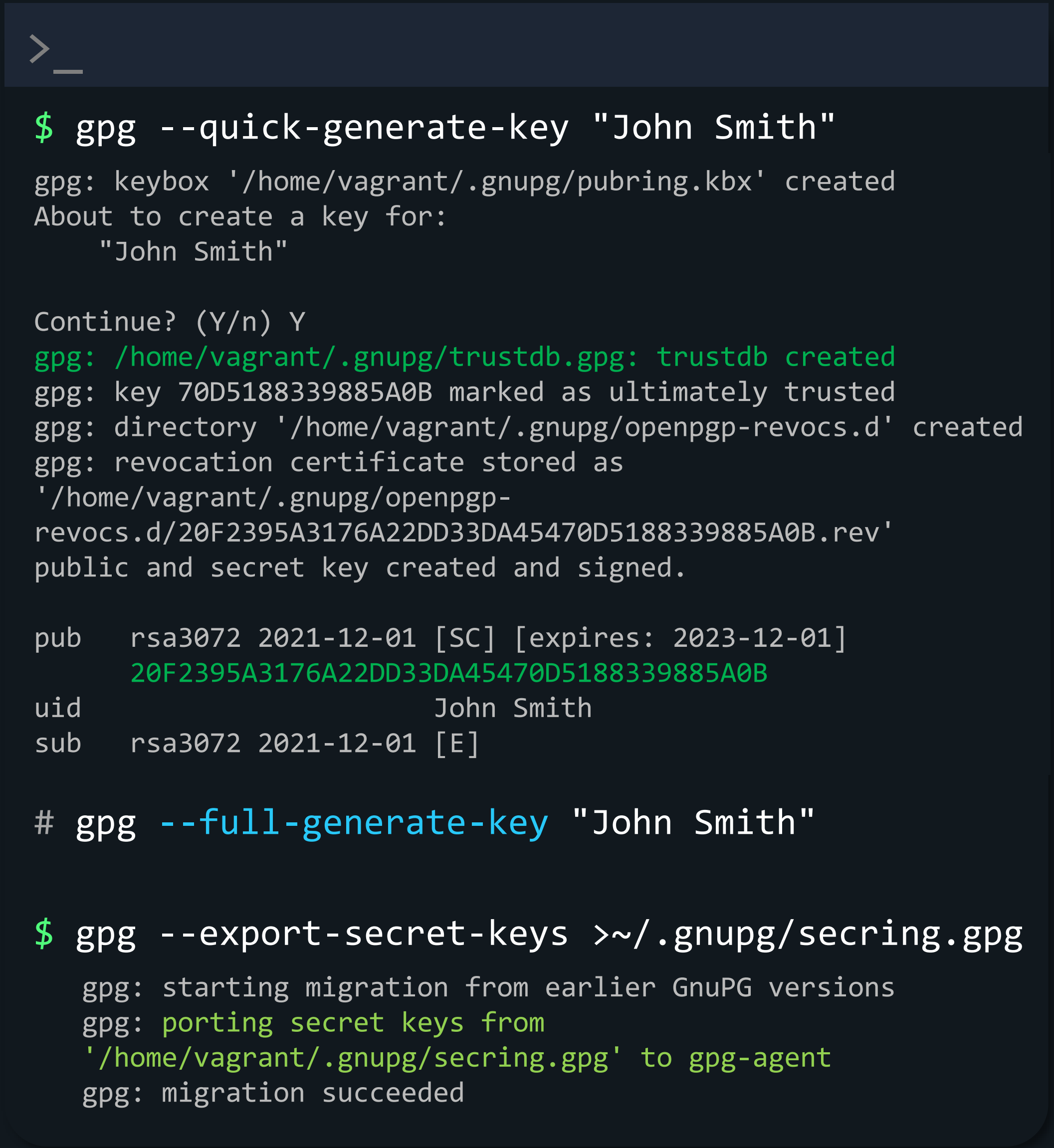
Helm Provenance and Integrity



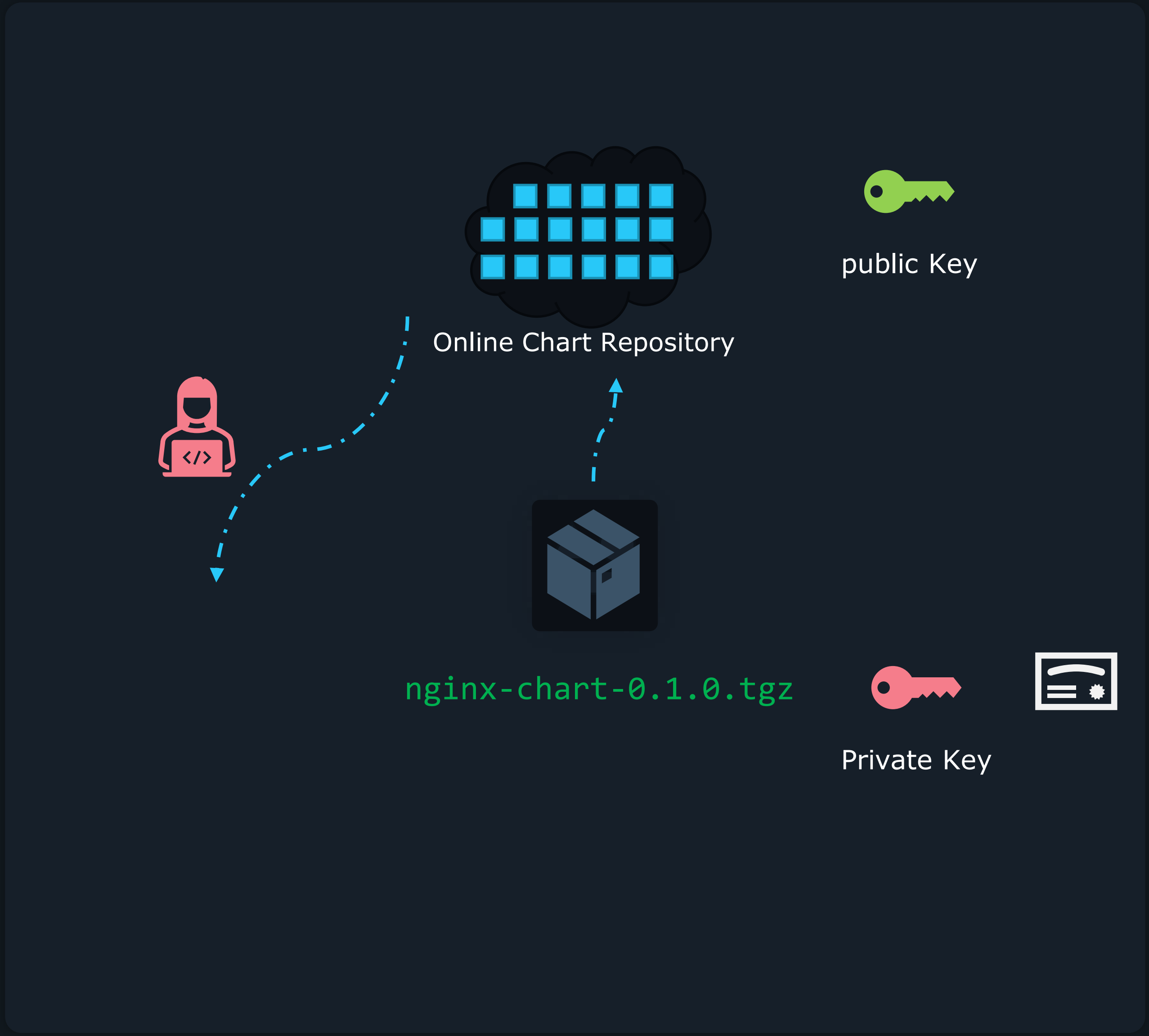
```
$ ls nginx-chart
  charts  Chart.yaml  templates  values.yaml README.md
  LICENSE


$ helm package ./nginx-chart
Successfully packaged chart and saved it to:
/vagrant/nginx-chart-0.1.0.tgz
```



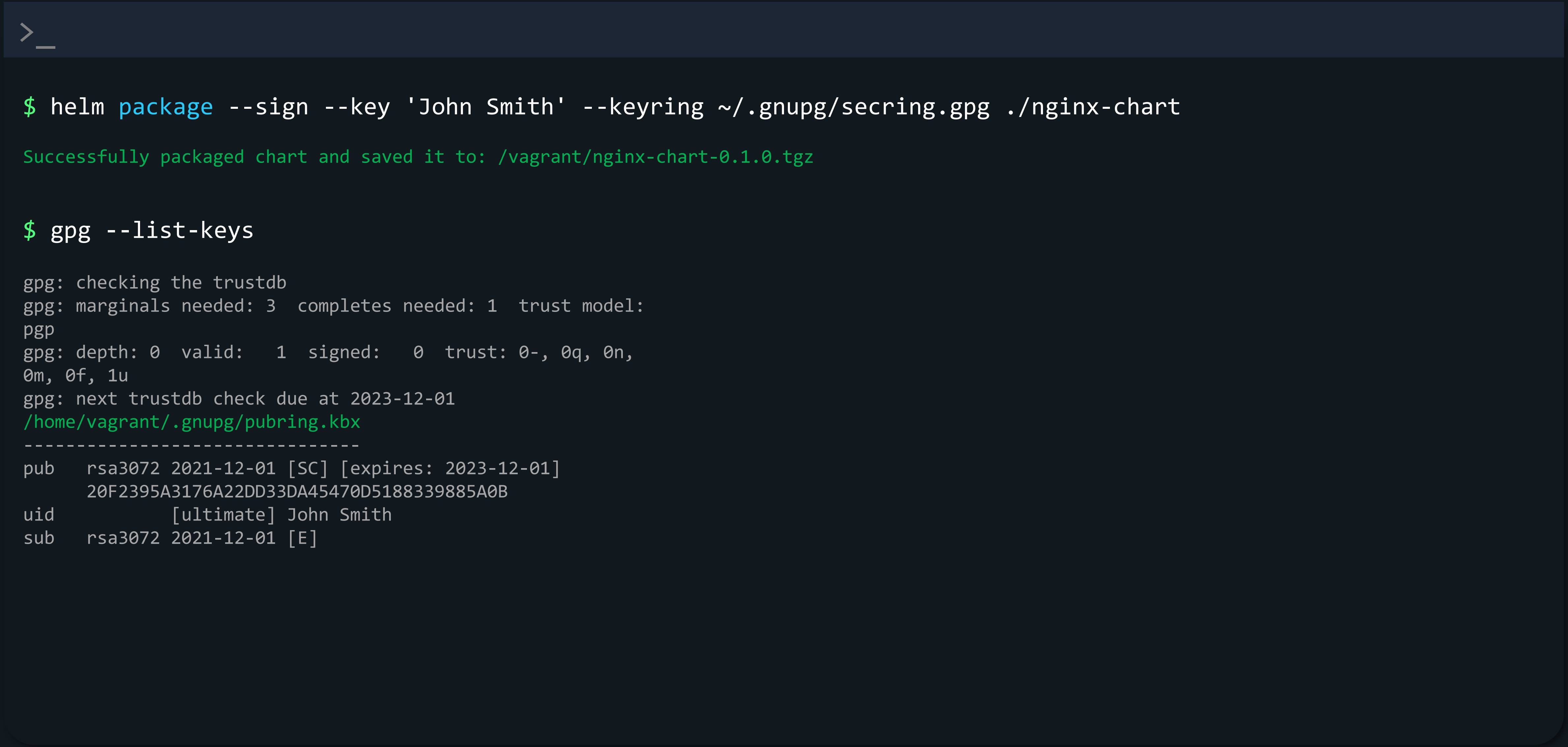
Packaging



```
$ gpg --quick-generate-key "John Smith"
gpg: keybox '/home/vagrant/.gnupg/pubring.kbx' created
About to create a key for:
    "John Smith"

Continue? (Y/n) Y
gpg: /home/vagrant/.gnupg/trustdb.gpg: trustdb created
gpg: key 70D5188339885A0B marked as ultimately trusted
gpg: directory '/home/vagrant/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as
'/home/vagrant/.gnupg/openpgp-
revocs.d/20F2395A3176A22DD33DA45470D5188339885A0B.rev'
public and secret key created and signed.

pub   rsa3072 2021-12-01 [SC] [expires: 2023-12-01]
      20F2395A3176A22DD33DA45470D5188339885A0B
uid                      John Smith
sub   rsa3072 2021-12-01 [E]

# gpg --full-generate-key "John Smith"


$ gpg --export-secret-keys >~/.gnupg/secring.gpg
   gpg: starting migration from earlier GnuPG versions
   gpg: porting secret keys from
   '/home/vagrant/.gnupg/secring.gpg' to gpg-agent
   gpg: migration succeeded
```

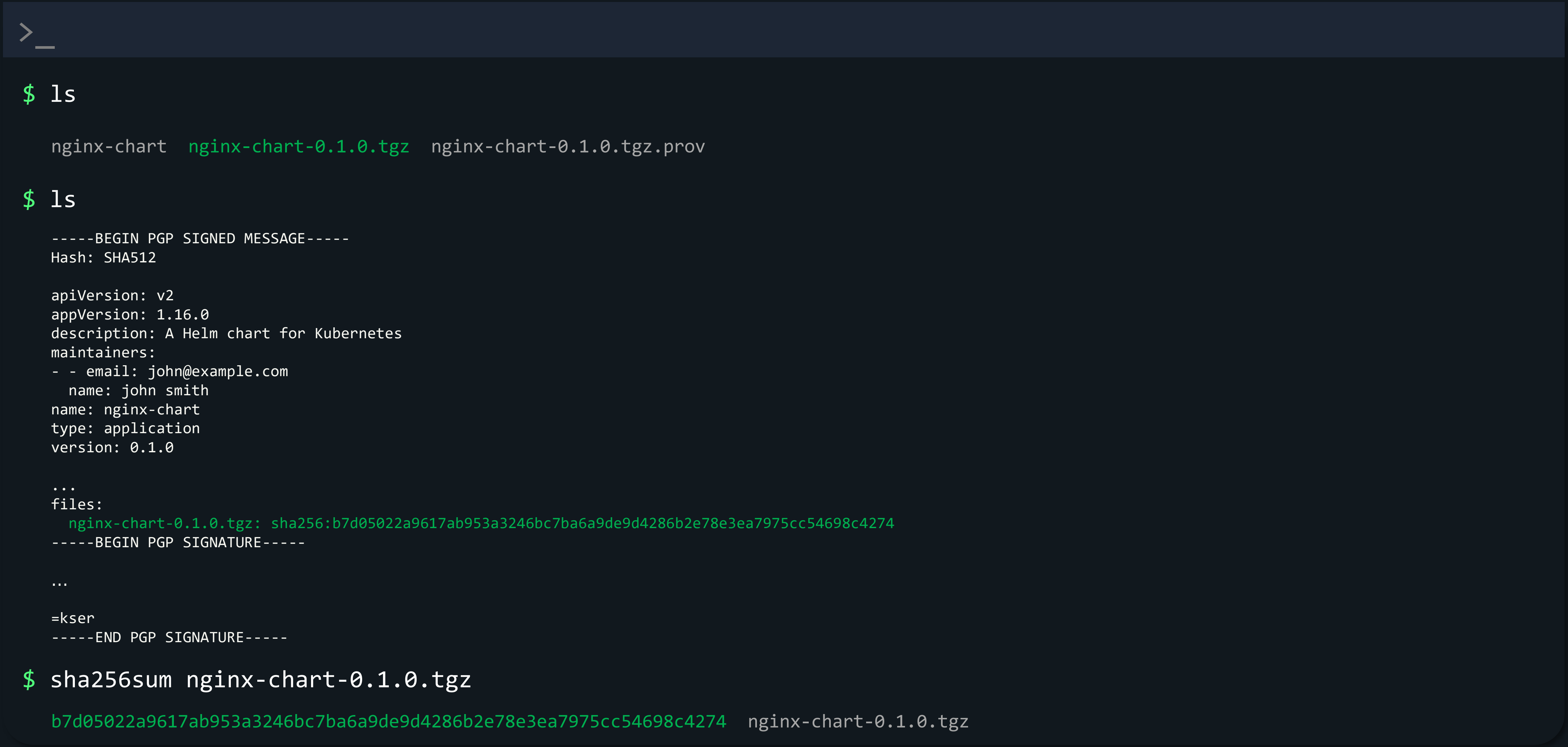


Signing and Verifying Helm Charts



```
$ helm package --sign --key 'John Smith' --keyring ~/.gnupg/secring.gpg ./nginx-chart

Successfully packaged chart and saved it to: /vagrant/nginx-chart-0.1.0.tgz


$ gpg --list-keys

gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model:
pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n,
0m, 0f, 1u
gpg: next trustdb check due at 2023-12-01
/home/vagrant/.gnupg/pubring.kbx
--------------------------------
pub   rsa3072 2021-12-01 [SC] [expires: 2023-12-01]
      20F2395A3176A22DD33DA45470D5188339885A0B
uid           [ultimate] John Smith
sub   rsa3072 2021-12-01 [E]
```

Provenance File



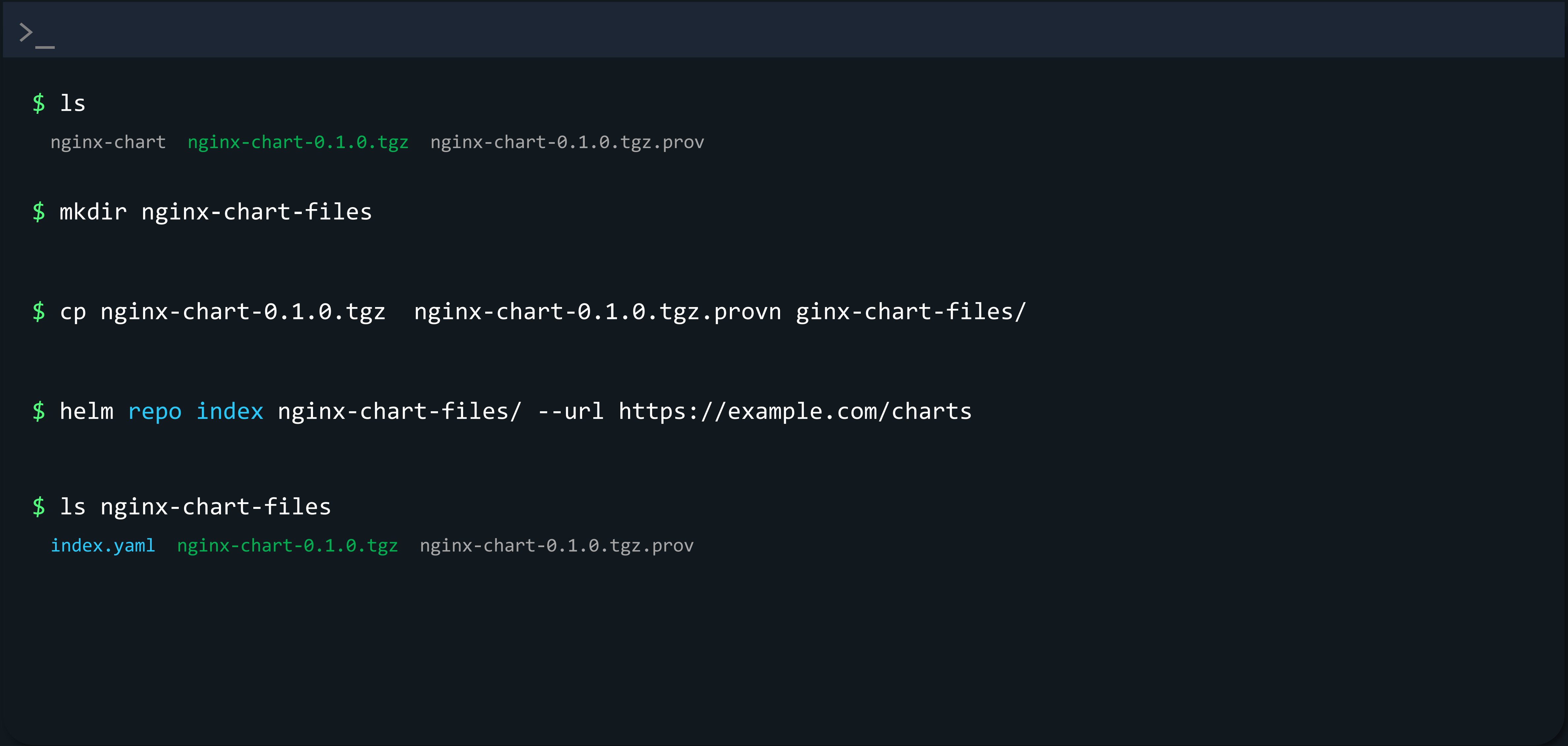
```
$ ls

  nginx-chart  nginx-chart-0.1.0.tgz  nginx-chart-0.1.0.tgz.prov

$ ls
  -----BEGIN PGP SIGNED MESSAGE-----
  Hash: SHA512

  apiVersion: v2
  appVersion: 1.16.0
  description: A Helm chart for Kubernetes
  maintainers:
  - - email: john@example.com
      name: john smith
  name: nginx-chart
  type: application
  version: 0.1.0

  ...
  files:
    nginx-chart-0.1.0.tgz: sha256:b7d05022a9617ab953a3246bc7ba6a9de9d4286b2e78e3ea7975cc54698c4274
  -----BEGIN PGP SIGNATURE-----

  …

  =kser
  -----END PGP SIGNATURE-----

$ sha256sum nginx-chart-0.1.0.tgz

  b7d05022a9617ab953a3246bc7ba6a9de9d4286b2e78e3ea7975cc54698c4274  nginx-chart-0.1.0.tgz
```

Verify Signature

```
>_

$ helm verify ./nginx-chart-0.1.0.tgz
  Error: failed to load keyring: open /home/vagrant/.gnupg/pubring.gpg: no such file or directory

$ gpg --export 'John Smith' > mypublickey


$ helm verify --keyring ./mypublickey ./nginx-0.1.0.tgz
  Signed by: John Smith
  Using Key With Fingerprint: 20F2395A3176A22DD33DA45470D5188339885A0B
  Chart Hash Verified: sha256:b7d05022a9617ab953a3246bc7ba6a9de9d4286b2e78e3ea7975cc54698c4274


$ gpg --recv-keys --keyserver keyserver.ubuntu.com 8D40FE0CACC3FED4AD1C217180BA57AAFAAD1CA5


$ helm install --verify nginx-chart-0.1.0
```

Uploading Charts

index.yaml

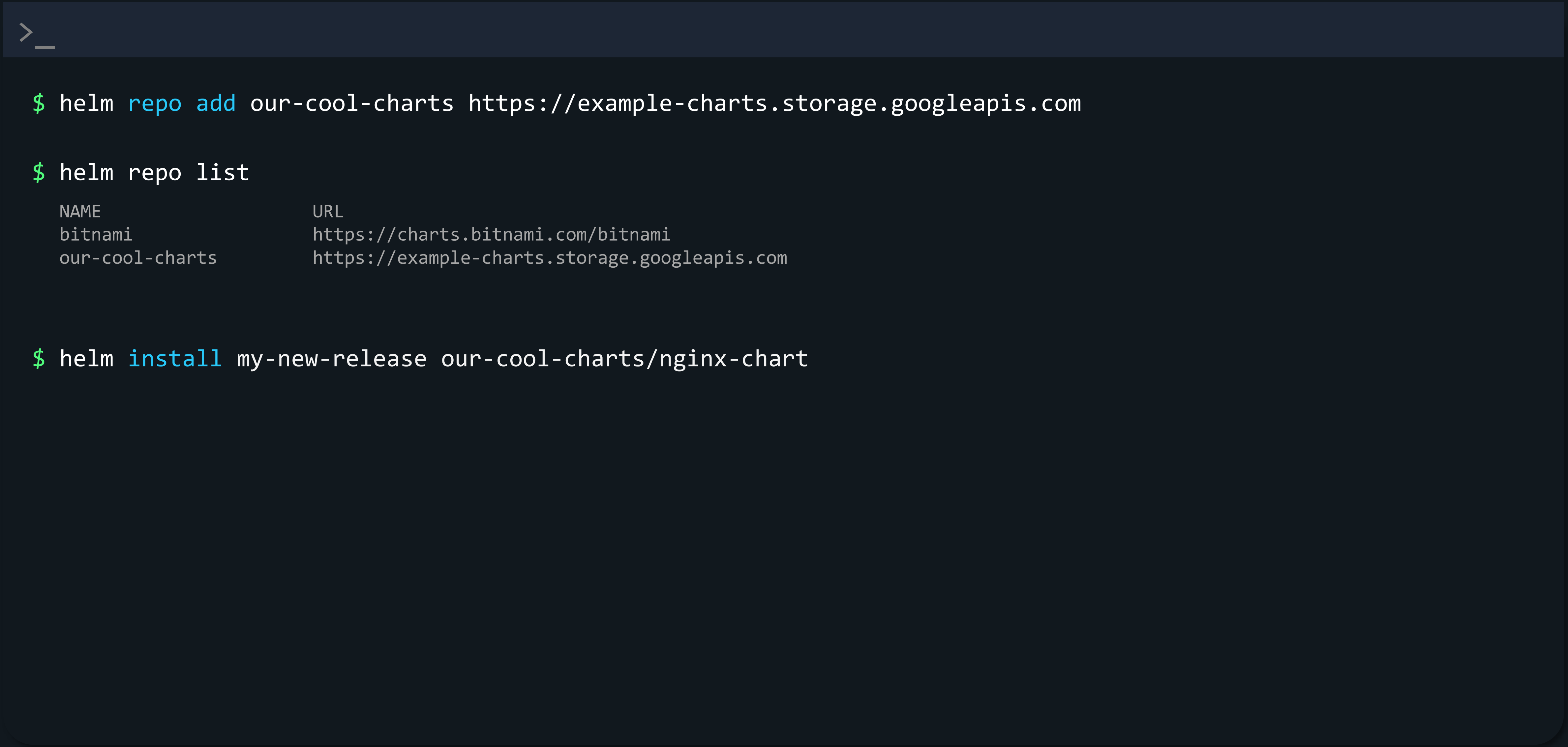


```
$ ls
  nginx-chart  nginx-chart-0.1.0.tgz  nginx-chart-0.1.0.tgz.prov

$ mkdir nginx-chart-files


$ cp nginx-chart-0.1.0.tgz  nginx-chart-0.1.0.tgz.provn ginx-chart-files/


$ helm repo index nginx-chart-files/ --url https://example.com/charts


$ ls nginx-chart-files
  index.yaml  nginx-chart-0.1.0.tgz  nginx-chart-0.1.0.tgz.prov
```

index.yaml

```yaml
apiVersion: v1
entries:
  nginx-chart:
  - apiVersion: v2
    appVersion: 1.16.0
    created: "2021-12-01T15:29:35.073405539Z"
    description: A Helm chart for Kubernetes
    digest: 2c83c29dc4c56d20c45c3de8eff521fbfb6ef6c0b66854a6f4b5539bebcff879
    maintainers:
    - email: john@example.com
      name: john smith
    name: nginx-chart
    type: application
    urls:
    - https://charts.bitnami.com/bitnami/nginx-chart-0.1.0.tgz
    version: 0.1.0
generated: "2021-12-01T15:29:35.047718855Z"
```

index.yaml



```
$ helm repo add our-cool-charts https://example-charts.storage.googleapis.com

$ helm repo list
  NAME                    URL
  bitnami                 https://charts.bitnami.com/bitnami
  our-cool-charts         https://example-charts.storage.googleapis.com


$ helm install my-new-release our-cool-charts/nginx-chart
```