**Kubernetes: Package Management with Helm**

1. **Install and Configure a Chart from the Artifact Hub**

**Check Helm version**

helm version --short

v3.18.4+gd80839c

**Get all commands of Helm**

helm

**List Helm charts**

helm list

NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION

**Show environment variables of Helm**

helm env

HELM\_BIN="helm"

HELM\_BURST\_LIMIT="100"

HELM\_CACHE\_HOME="/Users/ravishankarkushwaha/Library/Caches/helm"

HELM\_CONFIG\_HOME="/Users/ravishankarkushwaha/Library/Preferences/helm"

HELM\_DATA\_HOME="/Users/ravishankarkushwaha/Library/helm"

HELM\_DEBUG="false"

HELM\_KUBEAPISERVER=""

HELM\_KUBEASGROUPS=""

HELM\_KUBEASUSER=""

HELM\_KUBECAFILE=""

HELM\_KUBECONTEXT=""

HELM\_KUBEINSECURE\_SKIP\_TLS\_VERIFY="false"

HELM\_KUBETLS\_SERVER\_NAME=""

HELM\_KUBETOKEN=""

HELM\_MAX\_HISTORY="10"

HELM\_NAMESPACE="default"

HELM\_PLUGINS="/Users/ravishankarkushwaha/Library/helm/plugins"

HELM\_QPS="0.00"

HELM\_REGISTRY\_CONFIG="/Users/ravishankarkushwaha/Library/Preferences/helm/registry/config.json"

HELM\_REPOSITORY\_CACHE="/Users/ravishankarkushwaha/Library/Caches/helm/repository"

HELM\_REPOSITORY\_CONFIG="/Users/ravishankarkushwaha/Library/Preferences/helm/repositories.yaml"

**Explore the ArtifactHub:**

**Search cert-manager** in <https://artifacthub.io/> and filter by Official, Verified publishers and CNCF and Kind should be Helm charts

**Install the Chart**

kubectl apply -f <https://github.com/cert-manager/cert-manager/releases/download/v1.18.2/cert-manager.crds.yaml>

customresourcedefinition.apiextensions.k8s.io/certificaterequests.cert-manager.io created

customresourcedefinition.apiextensions.k8s.io/certificates.cert-manager.io created

customresourcedefinition.apiextensions.k8s.io/challenges.acme.cert-manager.io created

customresourcedefinition.apiextensions.k8s.io/clusterissuers.cert-manager.io created

customresourcedefinition.apiextensions.k8s.io/issuers.cert-manager.io created

customresourcedefinition.apiextensions.k8s.io/orders.acme.cert-manager.io created

**To install the chart with the release name cert-manager**

helm repo add jetstack https://charts.jetstack.io --force-update

"jetstack" has been added to your repositories

**Check that jetstack has been installed**

helm repo list

NAME URL

jetstack <https://charts.jetstack.io>

**Install the cert-manager helm chart**

helm install cert-manager --namespace cert-manager --version v1.18.2 jetstack/cert-manager

Error: INSTALLATION FAILED: create: failed to create: namespaces "cert-manager" not found

*“cert-manager” namespace not found, so create a namespace for cert-manager*

kubectl create ns cert-manager

namespace/cert-manager created

**Rerun command to install cert-manager helm chart**

helm install cert-manager --namespace cert-manager --version v1.18.2 jetstack/cert-manager

NAME: cert-manager

LAST DEPLOYED: Tue Jul 15 13:13:35 2025

NAMESPACE: cert-manager

STATUS: deployed

REVISION: 1

TEST SUITE: None

NOTES:

⚠️ WARNING: New default private key rotation policy for Certificate resources.

The default private key rotation policy for Certificate resources was

changed to `Always` in cert-manager >= v1.18.0.

Learn more in the [1.18 release notes](https://cert-manager.io/docs/releases/release-notes/release-notes-1.18).

cert-manager v1.18.2 has been deployed successfully!

In order to begin issuing certificates, you will need to set up a ClusterIssuer

or Issuer resource (for example, by creating a 'letsencrypt-staging' issuer).

More information on the different types of issuers and how to configure them

can be found in our documentation:

https://cert-manager.io/docs/configuration/

For information on how to configure cert-manager to automatically provision

Certificates for Ingress resources, take a look at the `ingress-shim`

documentation:

https://cert-manager.io/docs/usage/ingress/

**Get all from namespace cert-manager**

kubectl get all -n cert-manager

NAME READY STATUS RESTARTS AGE

pod/cert-manager-58dd99f969-p2x2l 1/1 Running 0 14m

pod/cert-manager-cainjector-55cd9f77b5-8hswt 1/1 Running 0 14m

pod/cert-manager-webhook-7987476d56-pklqd 1/1 Running 0 14m

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/cert-manager ClusterIP 10.109.249.110 <none> 9402/TCP 14m

service/cert-manager-cainjector ClusterIP 10.107.222.86 <none> 9402/TCP 14m

service/cert-manager-webhook ClusterIP 10.102.58.197 <none> 443/TCP,9402/TCP 14m

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/cert-manager 1/1 1 1 14m

deployment.apps/cert-manager-cainjector 1/1 1 1 14m

deployment.apps/cert-manager-webhook 1/1 1 1 14m

NAME DESIRED CURRENT READY AGE

replicaset.apps/cert-manager-58dd99f969 1 1 1 14m

replicaset.apps/cert-manager-cainjector-55cd9f77b5 1 1 1 14m

replicaset.apps/cert-manager-webhook-7987476d56 1 1 1 14m

**Get information about your chart with the Helm show command**

helm show chart jetstack/cert-manager

annotations:

artifacthub.io/category: security

artifacthub.io/license: Apache-2.0

artifacthub.io/prerelease: "false"

artifacthub.io/signKey: |

fingerprint: 1020CF3C033D4F35BAE1C19E1226061C665DF13E

url: https://cert-manager.io/public-keys/cert-manager-keyring-2021-09-20-1020CF3C033D4F35BAE1C19E1226061C665DF13E.gpg

apiVersion: v2

appVersion: v1.18.2

description: A Helm chart for cert-manager

home: https://cert-manager.io

icon: https://raw.githubusercontent.com/cert-manager/community/4d35a69437d21b76322157e6284be4cd64e6d2b7/logo/logo-small.png

keywords:

- cert-manager

- kube-lego

- letsencrypt

- tls

kubeVersion: '>= 1.22.0-0'

maintainers:

- email: cert-manager-maintainers@googlegroups.com

name: cert-manager-maintainers

url: https://cert-manager.io

name: cert-manager

sources:

- https://github.com/cert-manager/cert-manager

version: v1.18.2

**Show values of cert-manager using Helm chart**

helm show values jetstack/cert-manager

**This values file is too large so we will create a new file called values.yaml and redirect values in the newly created file.**

helm show values jetstack/cert-manager > values.yaml

**values.yaml file is created**

**Upgrade your Helm chart**

**See the version number of cert-manager**

helm ls -n cert-manager

NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION

cert-manager cert-manager 1 2025-07-15 13:13:35.951124 +0530 IST deployed cert-manager-v1.18.2 v1.18.2

**Helm upgrade**

helm upgrade

Error: "helm upgrade" requires 2 arguments

Usage: helm upgrade [RELEASE] [CHART] [flags]

**Helm upgrade cert-manager to 1.17.1**

helm upgrade cert-manager jetstack/cert-manager --version 1.17.1 -n cert-manager

Release "cert-manager" has been upgraded. Happy Helming!

NAME: cert-manager

LAST DEPLOYED: Tue Jul 15 15:21:50 2025

NAMESPACE: cert-manager

STATUS: deployed

REVISION: 2

TEST SUITE: None

NOTES:

cert-manager v1.17.1 has been deployed successfully!

In order to begin issuing certificates, you will need to set up a ClusterIssuer

or Issuer resource (for example, by creating a 'letsencrypt-staging' issuer).

More information on the different types of issuers and how to configure them

can be found in our documentation:

https://cert-manager.io/docs/configuration/

For information on how to configure cert-manager to automatically provision

Certificates for Ingress resources, take a look at the `ingress-shim`

documentation:

https://cert-manager.io/docs/usage/ingress/

**Get all from cert-manager namespace after upgrade**

kubectl get all -n cert-manager

NAME READY STATUS RESTARTS AGE

pod/cert-manager-cainjector-686546c9f7-v5b74 1/1 Running 0 82s

pod/cert-manager-d6746cf45-l4tm4 1/1 Running 0 82s

pod/cert-manager-webhook-5f79cd6f4b-qvn7l 1/1 Running 0 82s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/cert-manager ClusterIP 10.109.249.110 <none> 9402/TCP 129m

service/cert-manager-cainjector ClusterIP 10.107.222.86 <none> 9402/TCP 129m

service/cert-manager-webhook ClusterIP 10.102.58.197 <none> 443/TCP,9402/TCP 129m

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/cert-manager 1/1 1 1 129m

deployment.apps/cert-manager-cainjector 1/1 1 1 129m

deployment.apps/cert-manager-webhook 1/1 1 1 129m

NAME DESIRED CURRENT READY AGE

replicaset.apps/cert-manager-58dd99f969 0 0 0 129m

replicaset.apps/cert-manager-cainjector-55cd9f77b5 0 0 0 129m

replicaset.apps/cert-manager-cainjector-686546c9f7 1 1 1 82s

replicaset.apps/cert-manager-d6746cf45 1 1 1 82s

replicaset.apps/cert-manager-webhook-5f79cd6f4b 1 1 1 82s

replicaset.apps/cert-manager-webhook-7987476d56 0 0 0 129m

**Chapter 2 Challenge**

This is your first challenge of this course, and it will take you up to 20 minutes to complete.

The goal of this challenge is for you to install the prometheus-community hosted kube-prometheus-stack Helm chart in your Kubernetes cluster. You can find this chart in the Artifact Hub.

The release name should be “my-kps,” and you will install the chart in a new namespace called “challenge” that you should create with the command kubectl create ns challenge.

Success means that you have found the kube-prometheus-stack chart in the Artifact Hub, used a Helm command to install it in the challenge namespace, and when you run kubectl get all -n challenge you see many Kubernetes objects that have “my-kps” in their name.

I will show you my solution to this challenge in the next video. Good luck!

**Solution:**

**Create challenge namespace**

kubectl create ns challenge

namespace/challenge created

**Get all namespaces**

kubectl get ns

NAME STATUS AGE

cert-manager Active 137m

challenge Active 13s

default Active 20h

kube-node-lease Active 20h

kube-public Active 20h

kube-system Active 20h

**Get Helm repo of kube-prometheus-stack**

helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

helm repo update

"prometheus-community" has been added to your repositories

Hang tight while we grab the latest from your chart repositories...

...Successfully got an update from the "jetstack" chart repository

...Successfully got an update from the "prometheus-community" chart repository

Update Complete. ⎈Happy Helming!⎈

**Install kube-prometheus-stack with release name my-kps**

helm install my-kps prometheus-community/kube-prometheus-stack -n challenge

NAME: my-kps

LAST DEPLOYED: Tue Jul 15 15:32:07 2025

NAMESPACE: challenge

STATUS: deployed

REVISION: 1

NOTES:

kube-prometheus-stack has been installed. Check its status by running:

kubectl --namespace challenge get pods -l "release=my-kps"

Get Grafana 'admin' user password by running:

kubectl --namespace challenge get secrets my-kps-grafana -o jsonpath="{.data.admin-password}" | base64 -d ; echo

Access Grafana local instance:

export POD\_NAME=$(kubectl --namespace challenge get pod -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=my-kps" -oname)

kubectl --namespace challenge port-forward $POD\_NAME 3000

Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Prometheus instances using the Operator.

**Show all from namespace challenge**

kubectl get all -n challenge

NAME READY STATUS RESTARTS AGE

pod/alertmanager-my-kps-kube-prometheus-sta-alertmanager-0 0/2 Init:0/1 0 18s

pod/my-kps-grafana-7d95f57b-r9txb 0/3 ContainerCreating 0 47s

pod/my-kps-kube-prometheus-sta-operator-cc646486-ffmqj 1/1 Running 0 47s

pod/my-kps-kube-state-metrics-7db4559956-p76xt 1/1 Running 0 47s

pod/my-kps-prometheus-node-exporter-jxqj9 1/1 Running 0 47s

pod/prometheus-my-kps-kube-prometheus-sta-prometheus-0 0/2 Init:0/1 0 17s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/alertmanager-operated ClusterIP None <none> 9093/TCP,9094/TCP,9094/UDP 18s

service/my-kps-grafana ClusterIP 10.102.34.176 <none> 80/TCP 47s

service/my-kps-kube-prometheus-sta-alertmanager ClusterIP 10.99.149.62 <none> 9093/TCP,8080/TCP 47s

service/my-kps-kube-prometheus-sta-operator ClusterIP 10.108.246.206 <none> 443/TCP 47s

service/my-kps-kube-prometheus-sta-prometheus ClusterIP 10.111.243.124 <none> 9090/TCP,8080/TCP 47s

service/my-kps-kube-state-metrics ClusterIP 10.96.210.32 <none> 8080/TCP 47s

service/my-kps-prometheus-node-exporter ClusterIP 10.96.6.92 <none> 9100/TCP 47s

service/prometheus-operated ClusterIP None <none> 9090/TCP 17s

NAME DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR AGE

daemonset.apps/my-kps-prometheus-node-exporter 1 1 1 1 1 kubernetes.io/os=linux 47s

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/my-kps-grafana 0/1 1 0 47s

deployment.apps/my-kps-kube-prometheus-sta-operator 1/1 1 1 47s

deployment.apps/my-kps-kube-state-metrics 1/1 1 1 47s

NAME DESIRED CURRENT READY AGE

replicaset.apps/my-kps-grafana-7d95f57b 1 1 0 47s

replicaset.apps/my-kps-kube-prometheus-sta-operator-cc646486 1 1 1 47s

replicaset.apps/my-kps-kube-state-metrics-7db4559956 1 1 1 47s

NAME READY AGE

statefulset.apps/alertmanager-my-kps-kube-prometheus-sta-alertmanager 0/1 18s

statefulset.apps/prometheus-my-kps-kube-prometheus-sta-prometheus 0/1 17s

**Uninstall my-kps from challenge namespace**

helm uninstall my-kps -n challenge

release "my-kps" uninstalled

**Check namespace challenge**

kubectl get all -n challenge

No resources found in challenge namespace.

**Remove repo – Prometheus-community**

helm repo remove prometheus-community

"prometheus-community" has been removed from your repositories

1. **Deploy a Custom Application Using Helm**

**Create a new Helm chart from the command line**

helm create first-chart

Creating first-chart

**Check what Helm has created:**

ls first-chart

Chart.yaml charts templates values.yaml

**Deploy and update Kubernetes ConfigMap via Helm**

1. Delete templates folder
2. Create new templates folder and configmap.yaml file inside the folder
3. Add below code in the configmap.yaml file

---

apiVersion: v1

kind: ConfigMap

metadata:

name: first-chart-configmap

data:

port: "8080"

1. Now navigate into first-chart folder and run below command:

helm install first-chart .

NAME: first-chart

LAST DEPLOYED: Tue Jul 15 16:49:11 2025

NAMESPACE: default

STATUS: deployed

REVISION: 1

TEST SUITE: None

1. Check configmap namespace cm for newly installed first-chart:

kubectl get cm

NAME DATA AGE

first-chart-configmap 1 111s

kube-root-ca.crt 1 22h

1. Describe first-chart-configmap:

kubectl describe cm first-chart-configmap

Name: first-chart-configmap

Namespace: default

Labels: app.kubernetes.io/managed-by=Helm

Annotations: meta.helm.sh/release-name: first-chart

meta.helm.sh/release-namespace: default

Data

====

port:

----

8080

BinaryData

====

Events: <none>

1. Add new value in configmap.yaml file

allowTesting: "true"

configmap.yaml file after change

---

apiVersion: v1

kind: ConfigMap

metadata:

name: first-chart-configmap

data:

port: "8080"

allowTesting: "true"

1. Upgrade first-chart using Helm

helm upgrade first-chart .

Release "first-chart" has been upgraded. Happy Helming!

NAME: first-chart

LAST DEPLOYED: Tue Jul 15 16:55:54 2025

NAMESPACE: default

STATUS: deployed

REVISION: 2

TEST SUITE: None

1. Describe first-chart-configmap after changes:

kubectl describe cm first-chart-configmap

Name: first-chart-configmap

Namespace: default

Labels: app.kubernetes.io/managed-by=Helm

Annotations: meta.helm.sh/release-name: first-chart

meta.helm.sh/release-namespace: default

Data

====

allowTesting:

----

true

port:

----

8080

BinaryData

====

Events: <none>

**Kubernetes secret using Helm**

1. Create secret.yaml file

---

apiVersion: v1

kind: Secret

metadata:

name: first-secret

type: Opaque

data:

username:

password:

---

# Convert to base64

username: 'admin'

password: '4w572$9sns1$!'

1. Convert username and password into base64 using below command:

echo -n 'admin' | base64

YWRtaW4=

echo -n '4w572$9sns1$!' | base64

NHc1NzIkOXNuczEkIQ==

1. Change in secret.yaml file

---

apiVersion: v1

kind: Secret

metadata:

name: first-secret

type: Opaque

data:

username: YWRtaW4=

password: NHc1NzIkOXNuczEkIQ==

1. Helm keeps track of changes:

helm template first-chart .

---

# Source: first-chart/templates/secret.yaml

apiVersion: v1

kind: Secret

metadata:

name: first-secret

type: Opaque

data:

username: YWRtaW4=

password: NHc1NzIkOXNuczEkIQ==

---

# Source: first-chart/templates/configmap.yaml

apiVersion: v1

kind: ConfigMap

metadata:

name: first-chart-configmap

data:

port: "8080"

allowTesting: "true"

1. Deploy secret using helm upgrade command:

helm upgrade first-chart .

Release "first-chart" has been upgraded. Happy Helming!

NAME: first-chart

LAST DEPLOYED: Tue Jul 15 17:28:02 2025

NAMESPACE: default

STATUS: deployed

REVISION: 3

TEST SUITE: None

1. Check secret created in Kubernetes. Get all secrets:

kubectl get secrets

NAME TYPE DATA AGE

first-secret Opaque 2 89s

sh.helm.release.v1.first-chart.v1 helm.sh/release.v1 1 40m

sh.helm.release.v1.first-chart.v2 helm.sh/release.v1 1 33m

sh.helm.release.v1.first-chart.v3 helm.sh/release.v1 1 89s

1. Describe first-secret

kubectl describe secret first-secret

Name: first-secret

Namespace: default

Labels: app.kubernetes.io/managed-by=Helm

Annotations: meta.helm.sh/release-name: first-chart

meta.helm.sh/release-namespace: default

Type: Opaque

Data

====

password: 13 bytes

username: 5 bytes

**Rollback a Helm release**

1. **Get all revisions**

helm history first-chart

REVISION UPDATED STATUS CHART APP VERSION DESCRIPTION

1 Tue Jul 15 16:49:11 2025 superseded first-chart-0.1.0 1.16.0 Install complete

2 Tue Jul 15 16:55:54 2025 superseded first-chart-0.1.0 1.16.0 Upgrade complete

3 Tue Jul 15 17:28:02 2025 deployed first-chart-0.1.0 1.16.0 Upgrade complete

1. **Roll back to most recent version:**

helm rollback first-chart

Rollback was a success! Happy Helming!

**Check history of first-chart to see rollback**

helm history first-chart

REVISION UPDATED STATUS CHART APP VERSION DESCRIPTION

1 Tue Jul 15 16:49:11 2025 superseded first-chart-0.1.0 1.16.0 Install complete

2 Tue Jul 15 16:55:54 2025 superseded first-chart-0.1.0 1.16.0 Upgrade complete

3 Tue Jul 15 17:28:02 2025 superseded first-chart-0.1.0 1.16.0 Upgrade complete

4 Tue Jul 15 21:41:50 2025 deployed first-chart-0.1.0 1.16.0 Rollback to 2

It has rollback to revision 2

1. **Rollback to long back revision to most recent version(In our case rollback to revision 1)**:

helm rollback first-chart 1

Rollback was a success! Happy Helming!

**Check history of first-chart to see rollback**

helm history first-chart

REVISION UPDATED STATUS CHART APP VERSION DESCRIPTION

1 Tue Jul 15 16:49:11 2025 superseded first-chart-0.1.0 1.16.0 Install complete

2 Tue Jul 15 16:55:54 2025 superseded first-chart-0.1.0 1.16.0 Upgrade complete

3 Tue Jul 15 17:28:02 2025 superseded first-chart-0.1.0 1.16.0 Upgrade complete

4 Tue Jul 15 21:41:50 2025 superseded first-chart-0.1.0 1.16.0 Rollback to 2

5 Tue Jul 15 21:45:19 2025 deployed first-chart-0.1.0 1.16.0 Rollback to 1

**Experiment with Helm’s Advanced Features**

**Render a ConfigMap value dynamically with Helm templating**

Template engine has to pull values from the Chart.yaml and values.yaml. To use it we have to use template directive {{ }}

1. Change ConfigMap name version in configmap.yaml:

---

apiVersion: v1

kind: ConfigMap

metadata:

name: first-chart-configmap-{{.Chart.Version}}

data:

port: "8080"

allowTesting: "true"

1. Check config map namespace:

kubectl get cm

NAME DATA AGE

first-chart-configmap 1 5h8m

kube-root-ca.crt 1 27h

1. Upgrade first-chart with newly updated configmap.yaml:

helm upgrade first-chart .

Release "first-chart" has been upgraded. Happy Helming!

NAME: first-chart

LAST DEPLOYED: Tue Jul 15 21:59:59 2025

NAMESPACE: default

STATUS: deployed

REVISION: 6

TEST SUITE: None

1. Now check whether name of configmap has changed or not:

kubectl get cm

NAME DATA AGE

first-chart-configmap-0.1.0 2 60s

kube-root-ca.crt 1 27h

configmap name has been changed as per value from Chart.yaml

1. Change in version in Chart.yaml file from 0.1.0 to 0.1.1, Now upgrade and check for updated name-version:

helm upgrade first-chart .

Release "first-chart" has been upgraded. Happy Helming!

NAME: first-chart

LAST DEPLOYED: Tue Jul 15 22:04:00 2025

NAMESPACE: default

STATUS: deployed

REVISION: 7

TEST SUITE: None

kubectl get cm

NAME DATA AGE

first-chart-configmap-0.1.1 2 15s

kube-root-ca.crt 1 27h

**Using the values.yaml file**

**Add environment data configuration for staging environment:**

---

staging:

sample-key: smaple-12345

**Added above code in the top of values.yaml file. Access this value in configmap.yaml file as {{ .values.yaml.staging.sample-key }}**

**Dynamically render a value with a Helm conditional statement**

**We want to allow testing if environment is staging and don’t allow testing if it is production**

**Add below condition in configmap.yaml and remove allowTesting: “true” from file.**

allowTesting: {{ if eq .Values.env "staging" }} "true" {{ else }} "false" {{ end }}

**After above changes in configmap.yaml:**

---

apiVersion: v1

kind: ConfigMap

metadata:

name: first-chart-configmap-{{.Chart.Version}}

data:

port: "8080"

allowTesting: {{ if eq .Values.env "staging" }} "true" {{ else }} "false" {{ end }}

**Test this change:**

helm upgrade first-chart .

Release "first-chart" has been upgraded. Happy Helming!

NAME: first-chart

LAST DEPLOYED: Wed Jul 16 12:07:52 2025

NAMESPACE: default

STATUS: deployed

REVISION: 8

TEST SUITE: None

**Environment is staging so that it is showing allowTesting: “true”**

helm template first-chart .

---

# Source: first-chart/templates/secret.yaml

apiVersion: v1

kind: Secret

metadata:

name: first-secret

type: Opaque

data:

username: YWRtaW4=

password: NHc1NzIkOXNuczEkIQ==

---

# Source: first-chart/templates/configmap.yaml

apiVersion: v1

kind: ConfigMap

metadata:

name: first-chart-configmap-0.1.1

data:

port: "8080"

allowTesting: "true"

**Now change the environment from staging to production:**

Value.yaml

env: "production"

staging:

sample-key: smaple-12345

helm upgrade first-chart .

Release "first-chart" has been upgraded. Happy Helming!

NAME: first-chart

LAST DEPLOYED: Wed Jul 16 12:12:25 2025

NAMESPACE: default

STATUS: deployed

REVISION: 9

TEST SUITE: None

**Now allowTesting: “false” because environment is set to production**

helm template first-chart .

---

# Source: first-chart/templates/secret.yaml

apiVersion: v1

kind: Secret

metadata:

name: first-secret

type: Opaque

data:

username: YWRtaW4=

password: NHc1NzIkOXNuczEkIQ==

---

# Source: first-chart/templates/configmap.yaml

apiVersion: v1

kind: ConfigMap

metadata:

name: first-chart-configmap-0.1.1

data:

port: "8080"

allowTesting: "false"

**Chapter 4 Challenge**

This is your second challenge of this course, and it will take you up to 10 minutes to complete.

The goal of this challenge is for you to write a Helm conditional statement that changes the port number in a ConfigMap based on the environment.

If the environment is staging, the port number should be “8080.” If the environment is production, the port number should be “3000.”

Success means that when the environment is set to staging in the values.yaml file and you run the command helm template, the port number is 8080. When the environment is set to production and you run the command helm template, the port number is 3000.

Solution:

port: {{ if eq .Values.env "staging" }} "8080" {{ else }} "3000" {{ end }}

Check below github repo for detailed changes:

<https://github.com/LinkedInLearning/kubernetes-package-management-with-helm-3380008>