**Helm Installation**

[Helm | Installing Helm](https://helm.sh/docs/intro/install/)

For a Linux based system, installation process is very simple. Make sure you install a version 3 or later.

I used below script for helm installation in Centos

[azureuser@bastion ~]$ curl -fsSL -o get\_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3

[azureuser@bastion ~]$ chmod 700 get\_helm.sh

[azureuser@bastion ~]$ ./get\_helm.sh

Helm v3.9.0 is already latest

[azureuser@bastion ~]$

**Project Creation**

Once helm is installed, the next step is to build a project.

Let’s say the application name we are building is upwork-payment-app

[azureuser@bastion ~]$ helm create upwork-payment-app

Creating upwork-payment-app

Helm “create” command will create sample manifests for you which you can modify based on your requirements.

[azureuser@bastion ~]$ cd upwork-payment-app/

[azureuser@bastion upwork-payment-app]$ tree

.

├── charts

├── Chart.yaml

├── templates

│   ├── deployment.yaml

│   ├── \_helpers.tpl

│   ├── hpa.yaml

│   ├── ingress.yaml

│   ├── NOTES.txt

│   ├── serviceaccount.yaml

│   ├── service.yaml

│   └── tests

│   └── test-connection.yaml

└── values.yaml

3 directories, 10 files

[azureuser@bastion upwork-payment-app]$

**Update Helm Charts based on requirements**

As you can see here, config maps and secrets yamls are missing in automatically generated files, we can add them based on our requirements.

We also need to update deployment and service account values, but instead of directly updating manaifests files, this time we will play with “values.yml” file. This file contains all the variable which are used in deployment and service manifest files.

Default values.yaml file

|  |
| --- |
| # Default values for upwork-payment-app.  # This is a YAML-formatted file.  # Declare variables to be passed into your templates.  replicaCount: 1  image:  repository: nginx  pullPolicy: IfNotPresent  # Overrides the image tag whose default is the chart appVersion.  tag: ""  imagePullSecrets: []  nameOverride: ""  fullnameOverride: ""  serviceAccount:  # Specifies whether a service account should be created  create: true  # Annotations to add to the service account  annotations: {}  # The name of the service account to use.  # If not set and create is true, a name is generated using the fullname template  name: "" |

We need to modify it based on our requirements

|  |
| --- |
| [azureuser@bastion upwork-payment-app]$ cat values.yaml  # Default values for upwork-payment-app.  # This is a YAML-formatted file.  # Declare variables to be passed into your templates.  replicaCount: 1  image:  repository: devadeelregistry.azurecr.io/adshafqatupwork  pullPolicy: IfNotPresent  # Overrides the image tag whose default is the chart appVersion.  tag: "8"  imagePullSecrets: []  nameOverride: ""  fullnameOverride: ""  serviceAccount:  # Specifies whether a service account should be created  create: true  # Annotations to add to the service account  annotations: {}  # The name of the service account to use.  # If not set and create is true, a name is generated using the fullname template  name: "upworkwebappsa"  podAnnotations: {}  podSecurityContext: {}  # fsGroup: 2000  securityContext: {}  # capabilities:  # drop:  # - ALL  # readOnlyRootFilesystem: true  # runAsNonRoot: true  # runAsUser: 1000  service:  type: ClusterIP  port: 8080  ingress:  enabled: false  className: ""  annotations: {}  # kubernetes.io/ingress.class: nginx  # kubernetes.io/tls-acme: "true"  hosts:  - host: chart-example.local  paths:  - path: /  pathType: ImplementationSpecific  tls: []  # - secretName: chart-example-tls  # hosts:  # - chart-example.local  resources: {}  # We usually recommend not to specify default resources and to leave this as a conscious  # choice for the user. This also increases chances charts run on environments with little  # resources, such as Minikube. If you do want to specify resources, uncomment the following  # lines, adjust them as necessary, and remove the curly braces after 'resources:'.  limits:  cpu: 100m  memory: 128Mi  requests:  cpu: 100m  memory: 128Mi  autoscaling:  enabled: false  minReplicas: 1  maxReplicas: 100  targetCPUUtilizationPercentage: 80  # targetMemoryUtilizationPercentage: 80  nodeSelector: {}  tolerations: []  affinity: {} |

Add volumes related details in deployment file. Change port number from 80 to 8080. Remove rediness and liveness probes.

Add config map and secret yaml files

**Deploy Application in Kubernetes using helm commands**

**Installation**

[azureuser@bastion upwork-payment-app]$ helm install upwork-payment-app . -n upwork-payment

NAME: upwork-payment-app

LAST DEPLOYED: Tue Jun 21 11:59:58 2022

NAMESPACE: default

STATUS: deployed

REVISION: 1

NOTES:

1. Get the application URL by running these commands:

export POD\_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=upwork-payment-app,app.kubernetes.io/instance=upwork-payment-app" -o jsonpath="{.items[0].metadata.name}")

export CONTAINER\_PORT=$(kubectl get pod --namespace default $POD\_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")

echo "Visit http://127.0.0.1:8080 to use your application"

kubectl --namespace default port-forward $POD\_NAME 8080:$CONTAINER\_PORT

[azureuser@bastion upwork-payment-app]$

**Uninstallation**

[azureuser@bastion upwork-payment-app]$ helm uninstall upwork-payment-app -n upwork-payment

release "upwork-payment-app" uninstalled