Template will validate our code predeployemnt .
We will validate our code .
Create helm chart .
Will template command we can get a picture what all configuration incuded in our release as whole .

```
ubuntu@ip-172-31-47-135:~$ helm template helmlint/
---
# Source: helmlint/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: release-name-helmlint
  labels:
    helm.sh/chart: helmlint-0.1.0
    app.kubernetes.io/name: helmlint
    app.kubernetes.io/instance: release-name
    app.kubernetes.io/version: "1.16.0"
    app.kubernetes.io/managed-by: Helm
---
# Source: helmlint/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: release-name-helmlint
  labels:
    helm.sh/chart: helmlint-0.1.0
    app.kubernetes.io/name: helmlint
    app.kubernetes.io/instance: release-name
    app.kubernetes.io/version: "1.16.0"
    app.kubernetes.io/managed-by: Helm
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app.kubernetes.io/name: helmlint
    app.kubernetes.io/instance: release-name
---
# Source: helmlint/templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: release-name-helmlint
  labels:
    helm.sh/chart: helmlint-0.1.0
    app.kubernetes.io/name: helmlint
    app.kubernetes.io/instance: release-name
    app.kubernetes.io/version: "1.16.0"
```
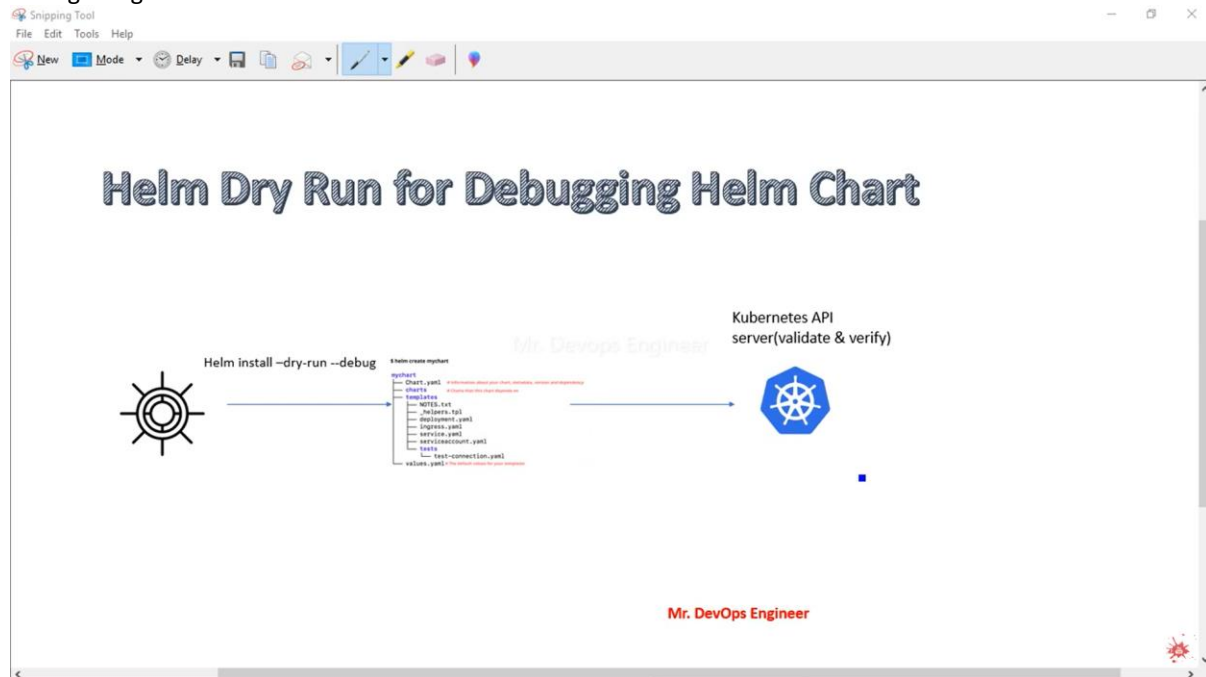
Lets make some mistake to check how helm template will help .
Comment kind in deployment.yaml file .
It will give error
helm template chart_name

Debug using helm :

Helm lint – for syntax error
Helm template – to validate chart values .
Using –dry-run we can de debug before actual deployment . it will give output by applying changes to Kubernetes ad dry-run .

To check all helm release present : helm ls -a

We will perform dryrun .

```
3 directories, 10 files
ubuntu@ip-172-31-47-135:~$ helm install newrelease --dry-run --debug hello-world/
install.go:194: [debug] Original chart version: ""
install.go:211: [debug] CHART PATH: /home/ubuntu/hello-world

NAME: newrelease
LAST DEPLOYED: Sat Apr  8 11:14:50 2023
NAMESPACE: default
STATUS: pending-install
REVISION: 1
USER-SUPPLIED VALUES:
{}

COMPUTED VALUES:
affinity: {}
autoscaling:
  enabled: false
```

helm list -a : to check all revisions .

helm delete release_name

kubectl get rs : to get all resources in current replicaset.

Upgarde helm :
We can upgrade our release using helm .

```
ubuntu@ip-172-31-47-135:~$ kubectl get rs
NAME                               DESIRED   CURRENT   READY   AGE
myrelase-hello-world-798b49845c    1         1         1       12s
ubuntu@ip-172-31-47-135:~$
```

Only 1 replicaset is running .

Increase replicaset to 2 in values.yaml file .

```
replicaCount: 2

image:
  repository: nginx
  pullPolicy: IfNotPresent
```

Lets perform upgrade for the new changes .
```
ubuntu@ip-172-31-47-135:~/hello-world$ helm upgrade myrelase .
Release "myrelase" has been upgraded. Happy Helming!
NAME: myrelase
LAST DEPLOYED: Sat Apr  8 11:38:56 2023
NAMESPACE: default
STATUS: deployed
REVISION: 2
NOTES:
1. Get the application URL by running these commands:
   export POD_NAME=$(kubectl get pods --namespace default -l "app.kuberne
   export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME
   echo "Visit http://127.0.0.1:8080 to use your application"
   kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_POR
```
Now release is upgraded and this is revision 2 .

Replicaset is increased to 2
```
   kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PO
ubuntu@ip-172-31-47-135:~/hello-world$ kubectl get rs
NAME                               DESIRED   CURRENT   READY   AGE
myrelase-hello-world-798b49845c    2         2         2       3m49s
ubuntu@ip-172-31-47-135:~/hello-world$
```

```
myrelase hello world 798b49845c   2        2        2       3m49s
ubuntu@ip-172-31-47-135:~/hello-world$ helm list -a
NAME        NAMESPACE    REVISION    UPDATED                                      STATUS      CHART             APP VERSION
myrelase    default      2           2023-04-08 11:38:56.022213564 +0000 UTC deployed        hello-world-0.1.0  1.16.0
ubuntu@ip-172-31-47-135:~/hello-world$
```

So we upgraded a release to new revision using helm chart .

Variables in helm chart :
Lets variabalize service type .

```
service:
  type: NodePort
  port: 80
podSecurityContext: {}
  # fsGroup: 2000


securityContext: {}
```
Add above in values.yaml file .

Update variable in service.yaml file .

```yaml
apiVersion: v1
kind: Service
metadata:
  name: {{ include "hello-world.fullname" . }}
  labels:
    {{- include "hello-world.labels" . | nindent 4 }}
spec:
  type: {{ .Values.service.type }}
  ports:
    - port: {{ .Values.service.port }}
      targetPort: http
      protocol: TCP
      name: http
  selector:
    {{- include "hello-world.selectorLabels" . | nindent 4 }}
~
~
~
```

Service is variabized .

So we have used variables in helm .

Helm rollback :
Rollback our release to previous version .

```
ubuntu@ip-172-31-47-135:~$ helm rollback myrelase 1
Rollback was a success! Happy Helming!
ubuntu@ip-172-31-47-135:~$
```

We rollback our release to older revision .

```
Rollback was a success! Happy Helming!
ubuntu@ip-172-31-47-135:~$ kubectl get all
NAME                                       READY   STATUS    RESTARTS   AGE
pod/myrelase-hello-world-798b49845c-vdw6d   1/1     Running   0          18m

NAME                         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/kubernetes           ClusterIP   10.96.0.1       <none>        443/TCP   107m
service/myrelase-hello-world ClusterIP   10.109.139.30   <none>        80/TCP    18m

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/myrelase-hello-world   1/1     1            1           18m

NAME                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/myrelase-hello-world-798b49845c   1         1         1       18m
ubuntu@ip-172-31-47-135:~$
```

Rs reduced to 1 pod .