# Refactoring Charts - Best practices

The DevOps team wants to refactor the umbrella chart so that it is better re-usable and can be shared with other teams as well. Currently, the database connection string needs to be provided in values file. A better approach would be to build the string by using go templates, with the help of functions. Also, the name of the kubernetes service, deployment & other objects can be constructed using functions.

## Learning Outcomes

After completing the lab, you will be able to understand

- Functions
- Scope with "with"
- Controlling Whitespaces & Indentations
- Logical Operators, Flow Control (conditions & loops)
- Variables
- Helper Functions and Sub Templates

## Template file

1. Create template files for both the `api` and `mysql` charts

   `pages/charts/api/templates/_helpers.tpl` & `pages/charts/mysql/templates/_helpers.tpl`

   ```
   touch ~/workspace/helm-charts/pages/charts/api/templates/_helpers.tpl
   touch ~/workspace/helm-charts/pages/charts/mysql/templates/_helpers.tpl
   ```

2. Define `api.fullname` and `api.getdbserviceurl` functions inside `api` chart

   `pages/charts/api/templates/_helpers.tpl`

```
{{- define "api.fullname" -}}
{{- if .Values.fullnameOverride -}}
{{- .Values.fullnameOverride -}}
{{- else -}}
{{- printf "%s-%s" .Release.Name .Chart.Name | trunc 63 | trimSuffix "-" -}}
{{- end -}}
{{- end -}}

{{- define "api.getdbserviceurl" -}}
{{- list "jdbc:mysql://" .Values.mysql_svc_name "/" .Values.dbname | join "" | q
uote -}}
{{- end -}}
```

3. Define `mysql.fullname` and `mysql.getdbserviceurl` functions inside `mysql` chart

`pages/charts/mysql/templates/_helpers.tpl`

```
{{- define "mysql.fullname" -}}
{{- if .Values.fullnameOverride -}}
{{- .Values.fullnameOverride -}}
{{- else -}}
{{- printf "%s-%s" .Release.Name .Chart.Name | trunc 63 | trimSuffix "-" -}}
{{- end -}}
{{- end -}}

{{- define "mysql.getdbserviceurl" -}}
{{- list "jdbc:mysql://" .Values.mysql_svc_name "/" .Values.env.MYSQL_DATABASE |
join "" | quote -}}
{{- end -}}
```

# Template and helper functions

1. Update `values.yaml` for both charts to externalize service and database names

`pages/charts/api/values.yaml`

```
mysql_svc_name: mysql
dbname: pages

deployment:
  containerPort: 8080
  replicaCount: 1
```

`pages/charts/mysql/values.yaml`

```
mysql_svc_name: mysql
```

## Update the manifest files for pages api service

1. Config map remains the same for pages api, since we want to ensure the configmap name to be the same as chart name, which is needed by the application.
2. Patch the name field for `pages/charts/api/templates/secret.yaml`

```
name: {{ include "api.fullname" . }}
```

3. Update `pages/charts/api/templates/service.yaml`

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: {{ include "api.fullname" . }}
  name: {{ include "api.fullname" . }}
spec:
  ports:
  - port: {{ .Values.service.port }}
    protocol: {{ .Values.service.protocol }}
```

```
    targetPort: {{ .Values.service.targetPort }}
  selector:
    app: {{ include "api.fullname" . }}
  type: {{ .Values.service.type }}
```

4. Update `pages/charts/api/templates/deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: {{ include "api.fullname" . }}
  name: {{ include "api.fullname" . }}
spec:
  replicas: {{ .Values.deployment.replicaCount}}
  selector:
    matchLabels:
      app: {{ include "api.fullname" . }}
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 2
     {{- if ge .Values.deployment.replicaCount 5.0 }}
      maxUnavailable: 2
     {{- else }}
      maxUnavailable: 0
     {{- end }}
  template:
    metadata:
      labels:
        app: {{ include "api.fullname" . }}
    spec:
      containers:
        - image: {{ .Values.image.repository }}:{{ .Values.image.tag }}
          name: {{ .Chart.Name }}
```

```
          imagePullPolicy: {{ .Values.imagePullPolicy }}
          ports:
            - containerPort: {{ .Values.deployment.containerPort }}
          env:
            - name: PAGE_CONTENT
              valueFrom:
                configMapKeyRef:
                  name: {{ .Chart.Name }}
                  key: PAGE_CONTENT
            - name: SPRING_DATASOURCE_URL
              value: {{ include "api.getdbserviceurl" . }}
            - name: SPRING_DATASOURCE_USERNAME
              value: {{ .Values.env.SPRING_DATASOURCE_USERNAME | quote}}
            - name: SPRING_DATASOURCE_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: {{ include "api.fullname" . }}
                  key: password
            {{- with .Values.env }}
            - name: DEBUG
              value: {{ .DEBUG | quote}}
            - name: LOGGING_FILE_NAME
              value: {{ .LOGGING_FILE_NAME | quote}}
            - name: LOGGING_LEVEL_ORG_SPRINGFRAMEWORK_WEB
              value: {{ .LOGGING_LEVEL_ORG_SPRINGFRAMEWORK_WEB }}
            - name: LOGGING_LEVEL_ROOT
              value: {{ .LOGGING_LEVEL_ROOT }}
            {{- end }}
            - name: MANAGEMENT_ENDPOINTS_WEB_EXPOSURE_INCLUDE
              value: "*"
          volumeMounts:
            - name: node-dir
              mountPath: /{{ .Release.Namespace }}
          readinessProbe:
            tcpSocket:
```

```
              port: {{ .Values.readinessProbe.tcpSocket.port }}
            initialDelaySeconds: {{ .Values.readinessProbe.initialDelaySeconds
 }}
            periodSeconds: {{ .Values.readinessProbe.periodSeconds }}
          livenessProbe:
            httpGet:
              path: {{ .Values.livenessProbe.httpGet.path }}
              port: {{ .Values.livenessProbe.httpGet.port }}
            initialDelaySeconds: {{ .Values.livenessProbe.initialDelaySeconds }}
            periodSeconds: {{ .Values.livenessProbe.periodSeconds }}
      volumes:
        - name: node-dir
          hostPath:
            path: /{{ .Release.Namespace }}
```

## Update the manifest files for mysql

1. Patch the name field for `pages/charts/mysql/templates/config.yaml`

```
name: {{ include "mysql.fullname" . }}
```

2. Patch the name field for `pages/charts/mysql/templates/secret.yaml`

```
name: {{ include "mysql.fullname" . }}
```

3. Update `pages/charts/mysql/templates/service.yaml`

```
apiVersion: v1
kind: Service
metadata:
  name: {{ .Values.mysql_svc_name }}
  labels:
    app: {{ include "mysql.fullname" . }}
spec:
```

```
  ports:
    - port: {{ .Values.service.port }}
  selector:
    app: {{ include "mysql.fullname" . }}
  type: {{ .Values.service.type }}
```

4. Patch the name field for `pages/charts/mysql/templates/storage-class.yaml`

```
name: {{ include "mysql.fullname" . }}-{{ .Release.Namespace }}
```

5. Update `pages/charts/mysql/templates/pv.yaml`

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: {{ include "mysql.fullname" . }}-{{ .Release.Namespace }}
  labels:
    type: local
spec:
  storageClassName: {{ include "mysql.fullname" . }}-{{ .Release.Namespace }}
  capacity:
    storage: {{ .Values.pv.capacity.storage }}
  accessModes:
    - {{ .Values.pv.accessMode }}
  hostPath:
    path: {{ .Values.pv.hostPath.path }}
```

6. Update `pages/charts/mysql/templates/pvc.yaml`

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: {{ include "mysql.fullname" . }}
spec:
```

```
  storageClassName: {{ include "mysql.fullname" . }}-{{ .Release.Namespace }}
  accessModes:
    - {{ .Values.pv.accessMode }}
  resources:
    requests:
      storage: {{ .Values.pvc.resources.requests.storage }}
```

7. Update `pages/charts/mysql/templates/flyway-job.yaml`

```
apiVersion: batch/v1
kind: Job
metadata:
  name: {{ include "mysql.fullname" . }}
  labels:
    app: {{ include "mysql.fullname" . }}
  annotations:
    "helm.sh/hook": post-install
    "helm.sh/hook-delete-policy": hook-succeeded
spec:
  backoffLimit: 16
  template:
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: {{ .Values.job.image.repository }}:{{ .Values.job.image.tag }}
          imagePullPolicy: {{ .Values.imagePullPolicy }}
          args:
            - info
            - migrate
            - info
          env:
            - name: FLYWAY_URL
              value: {{ include "mysql.getdbserviceurl" . }}
            - name: FLYWAY_USER
              value: {{ .Values.job.env.FLYWAY_USER }}
```

```
            - name: FLYWAY_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: {{ include "mysql.fullname" . }}
                  key: password
            - name: FLYWAY_PLACEHOLDER_REPLACEMENT
              value: {{ .Values.job.env.FLYWAY_PLACEHOLDER_REPLACEMENT | quote}}
            - name: FLYWAY_PLACEHOLDERS_USERNAME
              valueFrom:
                configMapKeyRef:
                  name: {{ include "mysql.fullname" . }}
                  key: spring.datasource.username
            - name: FLYWAY_PLACEHOLDERS_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: {{ include "mysql.fullname" . }}
                  key: password
          volumeMounts:
            - mountPath: /flyway/sql
              name: sql
      volumes:
        - name: sql
          configMap:
            name: {{ include "mysql.fullname" . }}
      restartPolicy: Never
```

8. Update `pages/charts/mysql/templates/deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "mysql.fullname" . }}
  labels:
    app: {{ include "mysql.fullname" . }}
spec:
```

```
  selector:
    matchLabels:
      app: {{ include "mysql.fullname" . }}
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: {{ include "mysql.fullname" . }}
    spec:
      containers:
        - image: {{ .Values.image.repository }}:{{ .Values.image.tag }}
          name: {{ .Chart.Name }}
          imagePullPolicy: {{ .Values.imagePullPolicy }}
          env:
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: {{ include "mysql.fullname" . }}
                  key: password
            - name: MYSQL_SERVICE_HOST
              value: {{ .Values.env.MYSQL_SERVICE_HOST | quote }}
            - name: MYSQL_SERVICE_PORT
              value: {{ .Values.env.MYSQL_SERVICE_PORT | quote }}
            - name: MYSQL_DATABASE
              value: {{ .Values.env.MYSQL_DATABASE | quote }}
          ports:
            - containerPort: {{ .Values.deployment.containerPort }}
          volumeMounts:
            - name: mysql-persistent-storage
              mountPath: {{ .Values.volumeMounts.mountPath }}
      volumes:
        - name: mysql-persistent-storage
          persistentVolumeClaim:
            claimName: {{ include "mysql.fullname" . }}
```

## Deploy using helm chart

1. Before installing the helm chart check if your namespace exists and set the kubectl context to point to the right namespace.

```
kubectl get  ns
kubectl config get-contexts
kubectl config set-context --current --namespace [name-of-your-team]-dev
```

2. Uninstall the previous app as we cannot upgrade

3. Install the umbrella chart for pages app

```
helm template pages
helm uninstall pagesapp
helm install pagesapp pages --dry-run --debug
helm install pagesapp pages -n [name-of-your-team]-dev
```

4. Verify the installation and deployment

```
helm list
kubectl get deploy pagesapp-api
kubectl get svc pagesapp-api
```

5. Port forward to connect to pages service running inside K8s from the local machine

```
kubectl port-forward svc/pagesapp-api 8080:8080
```

6. Test the pages application by performing CRUD operations using curl/postman. Refer Pages Curl Guide for testing.

## Add NOTES for chart users

1. It is a recommended best practice to add a `NOTES.txt` containing instructions for accessing the installed application. This file is customizable as per the chart user needs.

2. Before adding the notes, add a label to the api service `access=external` for easy accessibility of the service.

3. Create `NOTES.txt` file in the templates directory of the parent chart

```
touch ~/workspace/helm-charts/pages/charts/api/templates/NOTES.txt
```

4. Below is a sample text for our `NOTES.txt` , which is customizable according to application usability

   `~/workspace/helm-charts/pages/charts/api/templates/NOTES.txt`

```
Thank you for installing {{ .Chart.Name }}.

Your release is named {{ .Release.Name }}.

To learn more about the release, try:

helm status {{ .Release.Name }}
helm get all {{ .Release.Name }}

To access the application, try:

SVC=$(kubectl get svc -l access=external --namespace {{ .Release.Namespace }} -l
access=external | awk 'NR==2{print $1}')

kubectl --namespace {{ .Release.Namespace }} port-forward svc/$SVC 8080:8080
```

5. Re-install/update the umbrella chart for pages app

```
helm upgrade api pages -n dev
```

## Task Accomplished

Devops team was successful in refactoring the helm chart to be simple and reusable.