Sharing Chart data

The DevOps team wants to make use of global variables and share data between charts.

Learning Outcomes

After completing the lab, you will be able to understand

- Global variables
- Sharing Data between charts

Define Chart dependencies

1. Update the parent `Chart.yaml`

   `pages/Chart.yaml`

   ```
   apiVersion: v2
   name: pages
   description: A Helm chart for Pages Application
   type: application
   version: 1.0.0
   appVersion: "1.0"
   dependencies:
     - name: pages-api
       version: 1.x.x # version: ~1.0.0
       repository: http://localhost:8080
     - name: mysql
       version: 1.x.x # version: ^1.0.0
       repository: http://localhost:8080
   ```

Update versioning

1. Update the chart version for both `mysql` and `api`

   `pages/charts/mysql/Chart.yaml`

   ```
   apiVersion: v2
   name: mysql
   description: A Helm chart for MYSQL database
   type: application
   version: 1.0.0
   appVersion: "1.0"
   ```

   `pages/charts/api/Chart.yaml`

   ```
   apiVersion: v2
   name: pages-api
   description: A Helm chart for Pages API backend service
   type: application
   version: 1.0.0
   appVersion: "1.0"
   ```

Adding global data

1. Remove `mysql_svc_name` key-value pair from `api/values.yaml` and `mysql/values.yaml`

2. Add global data to `pages/values.yaml` parent chart

   ```
   global:
     mysql_svc_name: pages-mysql
   ```

3. Update the function `getdbserviceurl` defined in `api` chart - `api/templates/_helpers.tpl`

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

4. Update function `getdbserviceurl` to use the global value and create a new function `getdbservicename` which returns the service name in `mysql/templates/_helpers.tpl`

```
{{- define "mysql.getdbservicename" -}}
{{- .Values.global.mysql_svc_name -}}
{{- end -}}


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

5. Update mysql service name in `mysql/templates/service.yaml` which gets evaluated from the function `getdbservicename`

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

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.