Exercise 8.2: Working with Kustomize

1. Kustomize is a tool for customizing Kubernetes configurations. Its built into kubectl CLI.

```
student@cp:~$ kubectl kustomize --help
```

```
Build a set of KRM resources using a 'kustomization.yaml' file. The DIR argument must be a path to
↪  a directory
containing 'kustomization.yaml', or a git repository URL with a path suffix specifying same with
↪  respect to the
repository root. If DIR is omitted, '.' is assumed.

Examples:
  # Build the current working directory
  kubectl kustomize
...
<output_omitted>
```

2. Create a directory structure and copy the resource files in appropriate directory.

```
student@cp:~$ mkdir -p myapp/base myapp/overlays/dev myapp/overlays/prod
student@cp:~$ tree myapp
```

```
myapp
|--base
|-- overlays
    |-- dev
    |-- prod
```

3. Copy appropriate resource yaml files from the Solutions directory to directory structure created above.

```
student@cp:~$ cp /home/student/LFS258/SOLUTIONS/s_08/*.yaml-base myapp/base/
student@cp:~$ cp /home/student/LFS258/SOLUTIONS/s_08/*.yaml-dev myapp/overlays/dev
student@cp:~$ cp /home/student/LFS258/SOLUTIONS/s_08/*.yaml-prod myapp/overlays/prod
```

```
student@cp:~$ tree myapp
```

```
myapp
|-- base
|   |-- deployment.yaml-base
|   |-- kustomization.yaml-base
|   |-- service.yaml-base
|-- overlays
    |-- dev
    |   |-- deployment-patch.yaml-dev
    |   |-- kustomization.yaml-dev
    |   |-- service-patch.yaml-dev
    |-- prod
        |-- deployment-patch.yaml-prod
        |-- kustomization.yaml-prod
        |-- service-patch.yaml-prod
```

```
5 directories, 9 files
```

4. Rename the manifest files in the base directory.

```
student@cp:~$ cd myapp/base
student@cp:~/myapp/base$ for file in *.yaml-base; do mv "$file" "${file/-base/}"; done
student@cp:~/myapp/base$ ls *.yaml
```

```
deployment.yaml  kustomization.yaml  service.yaml
```

5. Rename the manifest files in the overlays/dev directory.

```
student@cp:~/myapp/base$ cd ../overlays/dev/
student@cp:~/myapp/overlays/dev$ for file in *.yaml-dev; do mv "$file" "${file/-dev/}"; done
student@cp:~/myapp/overlays/dev$ ls *.yaml
```

```
deployment.yaml  kustomization.yaml  service.yaml
```

6. Rename the manifest files in the overlays/prod directory.

```
student@cp:~/myapp/overlays/dev$ cd ../prod
student@cp:~/myapp/overlays/prod$ for file in *.yaml-prod; do mv "$file" "${file/-prod/}"; done
student@cp:~/myapp/overlays/prod$ ls *.yaml
```

```
deployment.yaml  kustomization.yaml  service.yaml
```

7. Verify to see if the files and directory structure match the below output.

```
student@cp:~/myapp/overlays/prod$ cd
student@cp:~$ tree myapp
```

```
myapp
|-- base
|   |-- deployment.yaml
|   |-- kustomization.yaml
|   |-- service.yaml
|-- overlays
    |-- dev
    |   |-- deployment-patch.yaml
    |   |-- kustomization.yaml
    |   |-- service-patch.yaml
    |-- prod
        |-- deployment-patch.yaml
        |-- kustomization.yaml
        |-- service-patch.yaml

5 directories, 9 files
```

8. The kustomization.yaml manifest file in the base directory has the details of the resources needs tobe created along with additional metadata injection and name modification.

```
student@cp:~$ vim myapp/base/kustomization.yaml
```

```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
namePrefix: lf-
resources:
- deployment.yaml
- service.yaml
labels:
- includeSelectors: true
  pairs:
    company: linux-foundation


```

9. Build the configuration and preview the output before applying it to the cluster. The base resource configurations are used as a foundation, and then environment-specific modifications are applied when building for a particular environment.

```
student@cp:~$ kubectl kustomize myapp/base
student@cp:~$ kubectl kustomize myapp/overlays/dev
student@cp:~$ kubectl kustomize myapp/overlays/prod
```

10. Once Verified, the configuration can be applied to the cluster using the -k option along with the kubectl

```
student@cp:~$ kubectl apply -k myapp/base/
```

```
service/lf-myapp created
deployment.apps/lf-myapp created
```

11. Verify the resources have been deployed correctly and the metadata has been injected as per kustomization.yaml

```
student@cp:~$ kubectl get all -l company=linux-foundation
```

```
NAME                              READY   STATUS    RESTARTS   AGE
pod/lf-myapp-5b68c7d779-ngsq4     1/1     Running   0          3m23s
pod/lf-myapp-5b68c7d779-z8pth     1/1     Running   0          3m23s

NAME               TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/lf-myapp   ClusterIP   10.104.21.181   <none>        80/TCP    3m23s

NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/lf-myapp     2/2     2            2           3m23s

NAME                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/lf-myapp-5b68c7d779     2         2         2       3m23s
```

12. Likewise, the remaining configurations can also be patched

```
student@cp:~$ kubectl apply -k myapp/overlays/dev/
```

```
service/lf-myapp configured
deployment.apps/lf-myapp configured
```

13. Verify if the resources have been configured as per the manifest files present in the overlays/dev directory.
14. Clean up by deleting the resources.

```
student@cp:~$ kubectl delete  -k myapp/overlays/dev/
```

```
service "lf-myapp" deleted
deployment.apps "lf-myapp" deleted
```