1. Go to the `base` directory.
2. Open the `deployment.yaml` file and change the contents to be as follows:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: caddy-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: caddy
  template:
    metadata:
      labels:
        app: caddy
    spec:
      containers:
      - name: caddy
        image: caddy:alpine
        ports:
        - containerPort: 80
          name: http
        volumeMounts:
        - name: caddy-config
          mountPath: /etc/caddy/
          readOnly: true
      volumes:
      - name: caddy-config
        configMap:
          name: caddy-config
```

3. Create `Caddyfile` with the following contents:

```
:80
log {
    output stdout
    format json
}
root * /usr/share/caddy
file_server
```

4. Modify the `service.yaml` and paste the following:

```
apiVersion: v1
kind: Service
metadata:
  name: caddy-service
spec:
  selector:
    app: caddy
  ports:
  - name: http
    port: 80
```

5. Create a new file called `ingress.yaml` and add the following:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: caddy-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: example.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: caddy-service
            port:
              name: http
```

6. Modify the `kustomization.yaml` file to look as follows:

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

resources:
- deployment.yaml
- service.yaml
- ingress.yaml

configMapGenerator:
- name: caddy-config
  files:
  - Caddyfile

namespace: default
```

7. Move to the `overlays` directory.
8. Create a new file called `ingress.yaml` and add the following content:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: caddy-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  ingressClassName: nginx
  rules:
  - http:
      paths:
      - path: /caddy
        pathType: Prefix
        backend:
          service:
            name: caddy-service
            port:
              name: http
```

9. Modify the `kustomization.yaml` file to be as follows:

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

resources:
- ../base

namespace: default

patches:
- path: ingress.yaml
  target:
    kind: Ingress
    name: caddy-ingress
```

10. Create a new Argo CD application in the `argo-cd` directory called `caddy.yaml` and add the following:

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: caddy
  namespace: argocd
spec:
  project: default
```

```
  source:
    repoURL: 'https://gitlab.com/[username]/samplegitopsapp.git'
    path: overlays
    targetRevision: main
  destination:
    server: 'https://kubernetes.default.svc'
    namespace: default
  syncPolicy:
    automated:
      selfHeal: true
      prune: true
```

11. Create a new branch, add, commit, and push:

```
git checkout -b "feature/adds-caddy"
git add -A
git commit -m "Adds Caddy web server"
git push --set-upstream origin feature/adds-caddy
```

12. Click on the MR link, go to the MR, approve it and merge to master.
13. Go to the Argo CD UI and refresh the Argo CD app.
14. Go to the terminal and show the objects that were created:

```
kubectl get pods
kubectl get svc
kubectl get configmaps
kubectl get ing
```

15. Open a new browser window and navigate to `/caddy`.