**Nginx Reverse Proxy**Step 1: - For the Nginx Reverse Proxy, we need to deploy a file that contains the ConfigMap, workload, and service deployment.  
  
apiVersion: v1

kind: ConfigMap

metadata:

name: nginx-conf

data:

nginx.conf: |

user nginx;

worker\_processes 1;

events {

worker\_connections 10240;

}

http {

server {

listen 80;

server\_name localhost;

location / {

root /usr/share/nginx/html; #Change this line

index index.html index.htm;

}

location /\_\_/auth {

proxy\_pass https://httpbin.org;

}

}

}

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-reverse-proxy

spec:

selector:

matchLabels:

app: nginx-reverse-proxy

replicas: 1

template:

metadata:

labels:

app: nginx-reverse-proxy

spec:

containers:

- name: nginx-reverse-proxy

image: nginx

ports:

- containerPort: 80

volumeMounts:

- name: nginx-conf

mountPath: /etc/nginx/nginx.conf

subPath: nginx.conf

readOnly: true

volumes:

- name: nginx-conf

configMap:

name: nginx-conf

items:

- key: nginx.conf

path: nginx.conf

---

apiVersion: v1

kind: Service

metadata:

name: nginx-reverse-proxy

spec:

ports:

- port: 80

protocol: TCP

targetPort: 80

selector:

app: nginx-reverse-proxy  
  
  
  
  
  
Step 2: - In the proxy pass section, we need to add our Firebase URL for redirection.

proxy\_pass https://httpbin.org; (ex: https://bcbsm-dev.firebaseapp.com;)  
  
  
Step 3:- After making the required changes, we need to upload this config file to Cloud Shell and deploy this YAML file with the namespace:  
  
kubectl apply -f deploymentfile.yaml -n namespace-name  
  
Step 4:- Once the deployment is done, we need to check the config file, workload, and service deployed under the Kubernetes Engine section.

Step 5: - Cross-verify the URL in the proxy\_pass section of the config file, then add the service with the context path in the ingress backend configuration.  
  
      - backend:

          service:

            name: nginx-reverse-proxy

            port:

              number: 80

        path: /\_\_/auth/\*

        pathType: ImplementationSpecific

Step 6:- After adding into the ingress wait until it’s reflected the changes in backend services no need to add the path in health check section because this service of type cluster IP.