

Kubernetes Lab 6

1- Create Microservice called "redis-leader"

```
apiVersion: v1
kind: Namespace
metadata:
  name: dev
  labels:
    name: dev
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-leader
  namespace: dev
spec:
  replicas: 1
  selector:
    matchLabels:
      name: redis-leader
  template:
    metadata:
      labels:
        name: redis-leader
    spec:
      containers:
        - image: "docker.io/redis:6.0.5"
          name: redis-leader
          ports:
            - containerPort: 6379
```

```
apiVersion: v1
kind: Service
metadata:
  name: redis-leader
  namespace: dev
spec:
  selector:
    name: redis-leader
  ports:
    - protocol: TCP
      port: 6379
      targetPort: 6379
```

2- Create Microservice called "redis-follower"

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-follower
  namespace: dev
spec:
  replicas: 2
  selector:
    matchLabels:
      name: redis-follower
  template:
    metadata:
      labels:
        name: redis-follower
    spec:
      containers:
        - image: "gcr.io/google_samples/gb-redis-follower:v2"
          name: redis-follower
          ports:
            - containerPort: 6379
```

```
apiVersion: v1
kind: Service
metadata:
  name: redis-follower
  namespace: dev
spec:
  selector:
    name: redis-follower
  ports:
    - protocol: TCP
      port: 6379
      targetPort: 6379
```

3- Create Microservice called "frontend"

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
  namespace: dev
spec:
  replicas: 3
  selector:
    matchLabels:
      name: frontend
  template:
    metadata:
      labels:
        name: frontend
    spec:
      containers:
        - name: frontend
          image: "gcr.io/google_samples/gb-frontend:v5"
          env:
            - name: GET_HOSTS_FROM
              value: "dns"
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: frontend
  namespace: dev
spec:
  type: NodePort
  selector:
    name: frontend
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```

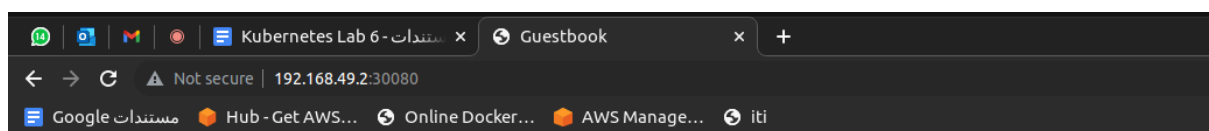
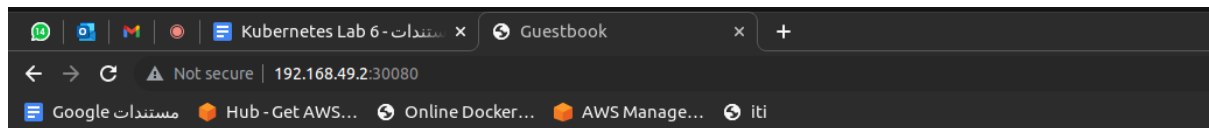
4- After creating the 3 microservice, you should be able to access the frontend service url and take screenshot of the output.

```
nathan@nathan-G3-3500:~/miniK8sProject$ kubectl get all --namespace=dev
NAME                                READY   STATUS    RESTARTS   AGE
pod/frontend-77c6589ccd-cf9z6      1/1     Running   0           22m
pod/frontend-77c6589ccd-fwf67      1/1     Running   0           20m
pod/frontend-77c6589ccd-m7vwc      1/1     Running   0           20m
pod/redis-follower-5f6c6d6664-bdzw8 1/1     Running   0           35m
pod/redis-follower-5f6c6d6664-k8fgj 1/1     Running   0           35m
pod/redis-leader-6885b5c6dd-bqf7d  1/1     Running   0           35m

NAME                                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/frontend                    NodePort      10.99.184.249   <none>           80:30080/TCP     35m
service/redis-follower               ClusterIP     10.97.231.221   <none>           6379/TCP         35m
service/redis-leader                 ClusterIP     10.104.74.15    <none>           6379/TCP         35m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/frontend            3/3     3             3           35m
deployment.apps/redis-follower      2/2     2             2           35m
deployment.apps/redis-leader        1/1     1             1           35m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/frontend-6fc678757c 0          0         0       35m
replicaset.apps/frontend-77c6589ccd  3          3         3       22m
replicaset.apps/redis-follower-5f6c6d6664 2          2         2       35m
replicaset.apps/redis-leader-6885b5c6dd 1          1         1       35m
nathan@nathan-G3-3500:~/miniK8sProject$
```



Hello World