rajiv@server-A:~/kubernetes/voting-app/voting-app$

Create the following file

```
-rw-r--r-- 1 rajiv rajiv  383 জুলাই       2 13:23 postgres-pod.yaml
-rw-r--r-- 1 rajiv rajiv  239 জুলাই       2 13:23 postgres-service.yaml
-rw-r--r-- 1 rajiv rajiv  231 জুলাই       2 13:23 redis-pod.yaml
-rw-r--r-- 1 rajiv rajiv  236 জুলাই       2 13:23 redis-service.yaml
-rw-r--r-- 1 rajiv rajiv  265 জুলাই       2 13:23 result-app-pod.yaml
-rw-r--r-- 1 rajiv rajiv  288 জুলাই       2 13:23 result-app-service.yaml
-rw-r--r-- 1 rajiv rajiv  273 জুলাই       2 13:23 voting-app-pod.yaml
-rw-r--r-- 1 rajiv rajiv  288 জুলাই       2 13:23 voting-app-service.yaml
-rw-r--r-- 1 rajiv rajiv  231 জুলাই       2 14:34 worker-app-pod.yaml
```

voting-app-pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: voting-app-pod
  labels:
    name: voting-app-pod
    app: demo-voting-app
spec:
  containers:
    - name: voting-app
      image: rajivsiddiqui/examplevotingapp_vote:v1
      ports:
        - containerPort: 80
```

voting-app-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: voting-service
  labels:
    name: voting-service
    app: demo-voting-app
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30004
  selector:
    name: voting-app-pod
    app: demo-voting-app
```

redis-pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: redis-pod
  labels:
    name: redis-pod
    app: demo-voting-app
spec:
  containers:
    - name: redis
      image: redis
      ports:
        - containerPort: 6379
```

redis-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: redis
  labels:
    name: redis-service
    app: demo-voting-app
spec:
  ports:
    - port: 6379
      targetPort: 6379
  selector:
    name: redis-pod
    app: demo-voting-app
```

postgres-pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: postgres-pod
  labels:
    name: postgres-pod
    app: demo-voting-app
spec:
```

```
  containers:
    - name: postgres
      image: postgres
      ports:
        - containerPort: 5432
      env:
        - name: POSTGRES_USER
          value: "postgres"
        - name: POSTGRES_PASSWORD
          value: "postgres"
```

postgres-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: db
  labels:
    name: postgres-service
    app: demo-voting-app
spec:
  ports:
    - port: 5432
      targetPort: 5432
  selector:
    name: postgres-pod
    app: demo-voting-app
```

result-app-pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: result-app-pod
  labels:
    name: result-app-pod
    app: demo-voting-app
spec:
  containers:
    - name: result-app
      image: rajivsiddiqui/examplevotingapp_result:v1
      ports:
        - containerPort: 80
```

result-app-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: result-service
  labels:
    name: result-service
    app: demo-voting-app
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30005
  selector:
    name: result-app-pod
    app: demo-voting-app
```

worker-app-pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: worker-app-pod
  labels:
    name: worker-app-pod
    app: demo-voting-app
spec:
  containers:
    - name: worker-app
      image: rajivsiddiqui/examplevotingapp_worker
```

#kubectl create -f voting-app-pod.yaml
#kubectl create -f voting-app-service.yaml
#kubectl get pods,svc

#minikube service voting-service --url

#kubectl create -f redis-pod.yaml
#kubectl create -f redis-service.yaml
#kubectl get pods,svc

#kubectl create -f postgres-pod.yaml
#kubectl create -f postgres-service.yaml
#kubeclt get pods,svc

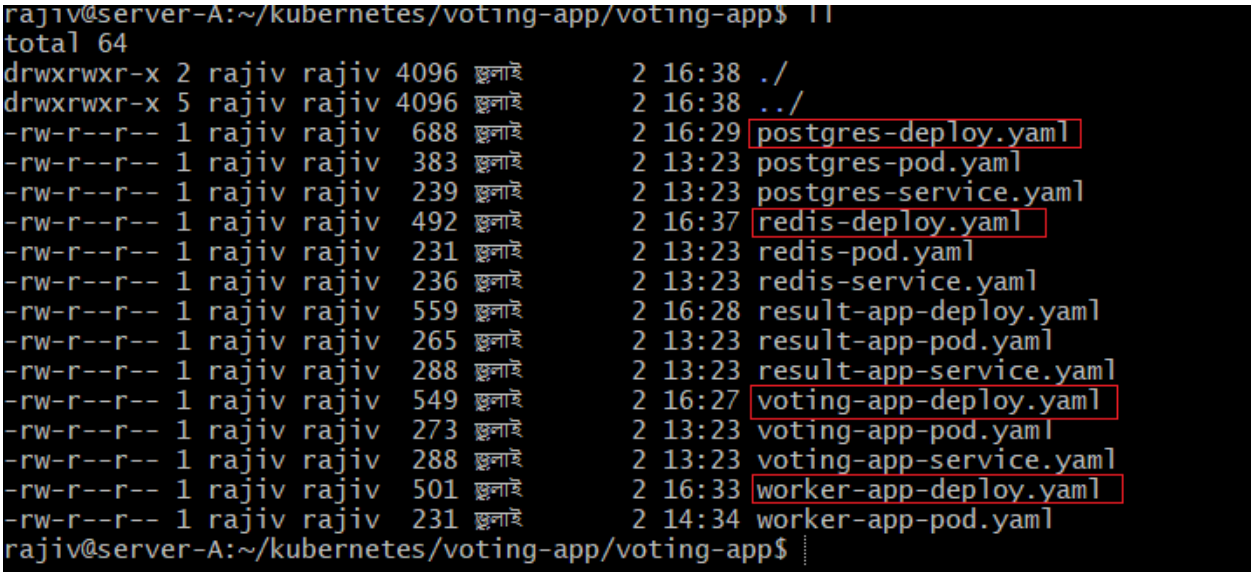
#kubectl create -f worker-app-pod.yaml
#kubectl get pods,svc

#kubectl create -f result-app-pod.yaml
#kubectl create -f result-app-service.yaml
#kubectl get pods,svc

Now brows from your ubuntu pc where the Kubernetes in installed

Now add deployment file

Now add 5 deployment file



voting-app-deploy.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: voting-app-deploy
  labels:
    name: voting-app-deploy
```

```
    app: demo-voting-app
spec:
  replicas: 1
  selector:
    matchLabels:
      name: voting-app-pod
      app: demo-voting-app

  template:
    metadata:
      name: voting-app-pod
      labels:
        name: voting-app-pod
        app: demo-voting-app
    spec:
      containers:
        - name: voting-app
          image: rajivsiddiqui/examplevotingapp_vote:v1
          ports:
            - containerPort: 80
```

redis-deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-deploy
  labels:
    name: redis-deploy
    app: demo-voting-app
spec:
  replicas: 1
  selector:
    matchLabels:
      name: redis-pod
      app: demo-voting-app

  template:
    metadata:
      name: redis-pod
      labels:
        name: redis-pod
        app: demo-voting-app
    spec:
```

```
      containers:
        - name: redis
          image: redis
          ports:
            - containerPort: 6379
```

postgres-deploy.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: postgres-deploy
  labels:
    name: postgres-deploy
    app: demo-voting-app
spec:
  replicas: 1
  selector:
    matchLabels:
      name: postgres-pod
      app: demo-voting-app

  template:
    metadata:
      name: postgres-pod
      labels:
        name: postgres-pod
        app: demo-voting-app
    spec:
      containers:
        - name: postgres
          image: postgres
          ports:
            - containerPort: 5432
          env:
            - name: POSTGRES_USER
              value: "postgres"
            - name: POSTGRES_PASSWORD
              value: "postgres"
```

result-app-deploy.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: result-app-deploy
  labels:
    name: result-app-deploy
    app: demo-voting-app
spec:
  replicas: 1
  selector:
    matchLabels:
      name: result-app-pod
      app: demo-voting-app

  template:
    metadata:
      name: result-app-pod
      labels:
        name: result-app-pod
        app: demo-voting-app
    spec:
      containers:
        - name: result-app
          image: rajivsiddiui/examplevotingapp_result:v1
          ports:
            - containerPort: 80
```

worker-app-deploy.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: worker-app-deploy
  labels:
    name: worker-app-deploy
    app: demo-voting-app
spec:
  replicas: 1
  selector:
    matchLabels:
      name: worker-app-pod
      app: demo-voting-app
```

```
  template:
    metadata:
      name: worker-app-pod
      labels:
        name: worker-app-pod
        app: demo-voting-app
    spec:
      containers:
        - name: worker-app
          image: rajivsiddiqui/examplevotingapp_worker
```

Command use in deployment:

#kubectl create -f voting-app-deploy.yaml

#kubectl create -f voting-app-service.yaml

#kubectl get pods,svc

#kubectl get deployment

#kubectl create -f redis-deploy.yaml

#kubectl create -f redis-service.yaml

#kubectl create -f postgres-deploy.yaml

#kubectl create -f postgres-service.yaml

#kubectl create -f result-app-deploy.yaml

#kubectl create -f result-app-service.yaml

#kubectl create -f worker-app-deploy.yaml

#kubectl get deployment

#kubectl get pods,svc

#kubectl get deployment,svc

#minikube service voting-service --url

#minikube service result-service --url

#kubectl get deployments voting-app-deploy

#kubectl scale deployment voting-app-deploy --replicas=3

#kubectl get deployments voting-app-deploy

#kubectl get pods

#kubectl scale deployment voting-app-deploy --replicas=1

#kubectl get pods

Voting app run on AWS EKS

EKS documentation file in AWS:
https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html

If we run the voting app to EKS then in the 2, service file we need to edit.

voting-app-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: voting-service
  labels:
    name: voting-service
    app: demo-voting-app
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
  selector:
    name: voting-app-pod
    app: demo-voting-app
```

result-app-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: result-service
  labels:
    name: result-service
    app: demo-voting-app
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
```

```
selector:
  name: result-app-pod
  app: demo-voting-app
```