

### Assignment -3

Assignment Date	12 September 2022
Student Name	Mr.Vemuru Neeraj Kumar
Student Roll Number	111519104165
Maximum Marks	2 Marks

#### Kubernetes:

##### dashboard-adminuser.yaml

apiVersion:

v1

```
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
```

---

```
apiVersion: v1
kind: Secret
metadata:
  name: admin-user-token
  namespace: kubernetes-dashboard
annotations:
  kubernetes.io/service-account.name: admin-user
type: kubernetes.io/service-account-token
```

---

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
```

##### flask\_deployment.yaml

apiVersion:

apps/v1

```
kind: Deployment
metadata:
  name: flask-app
```

```
spec:
  replicas: 3
  selector:
    matchLabels:
```

```
    app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
```

```
spec:
  containers:
    - name: webpage
      image: rajkiranss/flask
      imagePullPolicy: Never
    ports:
      - containerPort: 5000
        protocol: TCP
```

#### **flask\_ingress.yaml**

```
apiVersion:
  networking.k8s.io/v1
```

```
kind: Ingress
metadata:
  name: flask-app-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-redirect: "false"
```

```
spec:
  # ingressClassName: nginx
  rules:
    - http:
        paths:
          - backend:
              service:
                name: flask-app-service
                port:
                  number: 5000
              path: /
              pathType: Prefix
```

#### **flask\_service.yaml**

```
apiVersion:
  v1
```

```
kind: Service
metadata:
  name: flask-app-service
spec:
  type: ClusterIP
  ports:
    - port: 5000
  selector:
    app: flask-app
```

#### **ibm\_deployment.yaml**

```
apiVersion:
  apps/v1
```

```
kind: Deployment
metadata:
  name: flask-deploy
```

```
spec:
  replicas: 3
  selector:
    matchLabels:
      app: flask-deploy
  template:
    metadata:
      labels:
        app: flask-deploy
```

```
spec:
  containers:
    - name: web
      image: jp.icr.io/webpage/web
      imagePullPolicy: Always
      ports:
        - containerPort: 5000
          protocol: TCP
```

#### DOCKER FILE

```
FROM
python:3.10.4
```

```
WORKDIR /app
COPY requirements.txt ./
RUN pip install -r requirements.txt
COPY . .
EXPOSE 5000
CMD ["python", "./app.py"]
```

#### app.py

```
from flask import
Flask,
render_template,
request, url_for,
flash, redirect
```

```
app = Flask(__name__)
app.config['SECRET_KEY'] =
'df0331cefc6c2b9a5d0208a726a5d1c0fd37324feba25506'
```

```
@app.route('/create/', methods=('GET', 'POST'))
def create():
    if request.method == 'POST':
        title = request.form['title']
        content = request.form['content']
```

```
    if not title:
```

```

        flash('Title is required!')
    elif not content:
        flash('Content is required!')
    else:
        messages.append({'title': title, 'content': content})
        name = "Shyam Mohan"
        return redirect(url_for('index', messages=name ))

    return render_template('create.html')

messages = [{'title': 'Message One',
              'content': 'Message One Content'},
            {'title': 'Message Two',
              'content': 'Message Two Content'}
            ]

@app.route('/')
def index():
    return render_template('index.html', messages=messages)

@app.route('/admin')
def hello_admin():
    return 'Hello Admin'

@app.route('/guest/<guest>')
def hello_guest(guest):
    return 'Hello %s as Guest' % guest

@app.route('/user/<name>')
def hello_user(name):
    if name == 'admin':
        return redirect(url_for('hello_admin'))
    else:
        return redirect(url_for('hello_guest', guest = name))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)

```