Websphere Liberty : http + https | Docker & Kubernetes

```
→ git clone https://github.com/mgsgoms/vanakkam-world
→ cd vanakkam-world
→ mvn clean install
→ ls -lrt /home/srini_liberty/vanakkam-world/webapp/target/
```

```
→ mkdir ~/http
→ cd ~/http
→ cp ~/vanakkam-world/webapp/target/webapp.war .
→ vi Dockerfile
```

```
FROM icr.io/appcafe/websphere-liberty:kernel-java17-openj9-ubi
COPY --chown=1001:0 server.xml /config/
COPY --chown=1001:0 webapp.war /config/dropins/
RUN configure.sh
```

```
→ vi server.xml
```

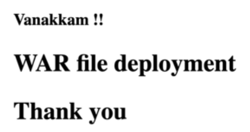
```
<server description="Liberty Base">
   <featureManager>
      <feature>servlet-4.0</feature>
   </featureManager>
   <httpEndpoint host="*" httpPort="9080" />
   <featureManager>
   <feature>adminCenter-1.0</feature>
</featureManager>

<administrator-role>
   <user>admin</user>
</administrator-role>

<basicRegistry>
   <user name="admin" password="adminpwd"/>
</basicRegistry>
</server>
```

```
→ sudo docker build -t liberty-base .
→ sudo docker ps
→ sudo docker run -d -p 8081:9080 liberty-base
→ sudo docker ps
```

Open the browser , get the GCP vm external ip and access with http://<ip>:8081



Vanakkam !!

WAR file deployment

Thank you

---

https:

→ mkdir ~/ssl
→ cd ssl
→ openssl req -newkey rsa:2048 -nodes -keyout key.key -x509 -days 365 -out cert.crt -subj "/CN=localhost"
→ openssl pkcs12 -export -in cert.crt -inkey key.key -out key.p12 -name default -password pass:redhat
→ ls -lrt
→ cp ~/http/webapp.war .
→ ls -lrt

→ vi Dockerfile

```
FROM icr.io/appcafe/websphere-liberty:kernel-java17-openj9-ubi
COPY --chown=1001:0 server.xml /config/
COPY --chown=1001:0 webapp.war /config/dropins/
COPY --chown=1001:0 key.p12 /config/key.p12
RUN configure.sh
```

→ vi server.xml

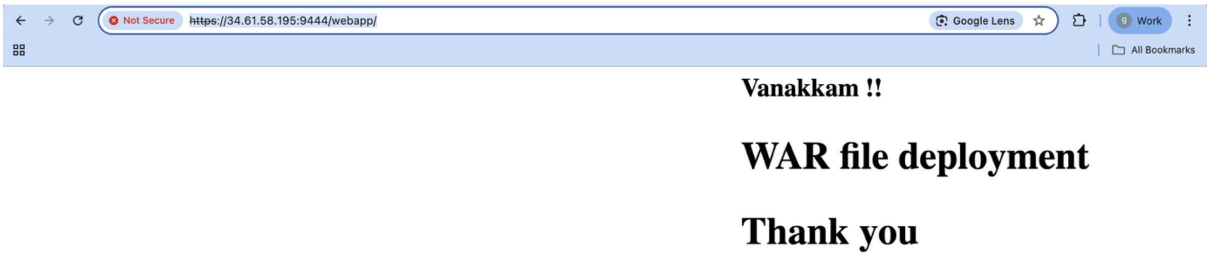
```
<server description="Liberty Base">
   <featureManager>
       <feature>servlet-4.0</feature>
   </featureManager>
     <!-- HTTP + HTTPS endpoint -->
   <httpEndpoint id="defaultHttpEndpoint"
              host="*"
              httpPort="9080"
              httpsPort="9443"
         keyStoreRef="defaultKeyStore" />
         <!-- Keystore configuration -->
   <keyStore id="defaultKeyStore" location="/config/key.p12" type="PKCS12"
password="redhat"/>
```

```
   <featureManager>
   <feature>adminCenter-1.0</feature>
</featureManager>

<administrator-role>
   <user>admin</user>
</administrator-role>

<basicRegistry>
   <user name="admin" password="adminpwd"/>
</basicRegistry>
</server>
```

→ sudo docker build -t liberty-https .
→ sudo docker run -d --name liberty-https -p 9082:9080 -p 9444:9443 liberty-https
→ sudo docker ps



Vanakkam !!

WAR file deployment

Thank you

Kubernetes

→ vi Dockerfile

```
# Dockerfile
FROM icr.io/appcafe/open-liberty:25.0.0.6-kernel-slim-java17-openj9-ubi

# Copy Liberty server configuration (without cert/key)
COPY --chown=1001:0 server.xml /config/

# Copy your WAR app
COPY --chown=1001:0 webapp.war /config/dropins/

# Configure Liberty to install features declared in server.xml
```

```
RUN configure.sh

EXPOSE 9080 9443
```

→ vi server.xml

```
<server description="Liberty Base HTTPS">
   <featureManager>
      <feature>servlet-4.0</feature>
      <feature>ssl-1.0</feature>
      <feature>adminCenter-1.0</feature>
   </featureManager>

   <httpEndpoint id="defaultHttpEndpoint"
               host="*"
               httpPort="9080"
               httpsPort="9443"
               keyStoreRef="defaultKeyStore"/>

   <!-- KeyStore will be mounted from Secret in Kubernetes -->
   <keyStore id="defaultKeyStore" location="/config/key.p12" type="PKCS12"
password="changeit"/>

   <administrator-role>
      <user>admin</user>
   </administrator-role>

   <basicRegistry>
      <user name="admin" password="adminpwd"/>
   </basicRegistry>
</server>
```

-→ docker build -t liberty-https:latest .
→ docker tag liberty-https:latest <your-registry>/liberty-https:latest
→ docker push <your-registry>/liberty-https:latest

Generate base64 of key.p12:

→ base64 -w 0 key.p12
→ vi server.xml

```
apiVersion: v1
kind: Secret
metadata:
  name: liberty-ssl
```

```
type: Opaque
data:
  key.p12: <BASE64_ENCODED_KEYP12>
```

→ vi configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: liberty-config
data:
  server.xml: |
    <server description="Liberty Base HTTPS">
        <featureManager>
            <feature>servlet-4.0</feature>
            <feature>ssl-1.0</feature>
            <feature>adminCenter-1.0</feature>
        </featureManager>

        <httpEndpoint id="defaultHttpEndpoint"
                      host="*"
                      httpPort="9080"
                      httpsPort="9443"
                      keyStoreRef="defaultKeyStore"/>

        <keyStore id="defaultKeyStore" location="/config/key.p12" type="PKCS12"
password="changeit"/>

        <administrator-role>
            <user>admin</user>
        </administrator-role>

        <basicRegistry>
            <user name="admin" password="adminpwd"/>
        </basicRegistry>
    </server>
```

→ vi deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: liberty-https
spec:
  replicas: 1
  selector:
    matchLabels:
```

```
   app: liberty-https
 template:
  metadata:
   labels:
    app: liberty-https
  spec:
   containers:
    - name: liberty
     image: cubensquare/liberty-https:latest
     ports:
      - containerPort: 9080
      - containerPort: 9443
     volumeMounts:
      # Mount ConfigMap as server.xml
      - name: config-volume
       mountPath: /config/server.xml
       subPath: server.xml
      # Mount Secret as key.p12
      - name: ssl-volume
       mountPath: /config/key.p12
       subPath: key.p12
   volumes:
    - name: config-volume
     configMap:
      name: liberty-config
    - name: ssl-volume
     secret:
      secretName: liberty-ssl
---
apiVersion: v1
kind: Service
metadata:
 name: liberty-https-svc
spec:
 selector:
  app: liberty-https
 ports:
  - port: 80
   targetPort: 9080
   name: http
  - port: 443
   targetPort: 9443
   name: https
 type: NodePort
```

→ kubectl apply -f secret.yaml

→ kubectl apply -f configmap.yaml
→ kubectl apply -f deployment.yaml

Validate the url with https://<external-ip>:nodeport