**Detailed Documentation for Deploying WSO2 API Manager on Kubernetes Using Helm and PostgreSQL**

**Overview**

This document provides a step-by-step guide to deploy WSO2 API Manager (WSO2 APIM) on a Kubernetes cluster using the official [WSO2 Helm chart](https://github.com/wso2/helm-apim/tree/main/all-in-one). The deployment integrates PostgreSQL as the database and includes configurations for Nginx as a reverse proxy.

**Key Components**

* **WSO2 API Manager**: API management solution deployed on Kubernetes.
* **PostgreSQL**: Database for storing WSO2 APIM data.
* **Nginx**: Acts as a reverse proxy for API Manager traffic.

**Prerequisites**

1. Kubernetes cluster (e.g., Minikube).
2. Helm 3 installed.
3. kubectl CLI installed and configured.
4. PostgreSQL client (psql) installed locally.
5. SSL certificates for Nginx.

**Step-by-Step Instructions**

**1. Clone the Helm Chart Repository**

Clone the WSO2 Helm chart repository:

git clone https://github.com/wso2/helm-apim.git

cd helm-apim/all-in-one

**2. Install and Configure Nginx**

Deploy Nginx as a reverse proxy:

**Install Nginx**

kubectl apply -f https://raw.githubusercontent.com/nginxinc/kubernetes-ingress/main/deploy/static/provider/kind/deploy.yaml

**Configure Nginx**

server {

listen 80;

server\_name localhost;

location / {

proxy\_pass http://localhost:9443;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

server {

listen 443 ssl;

server\_name localhost;

ssl\_certificate /path/to/certificate.crt;

ssl\_certificate\_key /path/to/private.key;

location / {

proxy\_pass http://localhost:9443;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

**3. Install PostgreSQL as a Pod and Expose PostgreSQL as a service**:

Create a PostgreSQL deployment:

kubectl run postgres --image=postgres:13 --env="POSTGRES\_USER=wso2carbon" --env="POSTGRES\_PASSWORD=wso2carbon" --env="POSTGRES\_DB=wso2am\_db" --port=5432

kubectl expose pod postgres --type=ClusterIP --port=5432 --target-port=5432

**4. Prepare and Run Database Scripts**

**Copy Database Scripts to Pod**

Copy the scripts from your local machine to the PostgreSQL pod:

kubectl cp /path/to/db/scripts.sql postgres:/tmp/postgresql.sql

**Execute the Scripts**

Access the PostgreSQL pod:

kubectl exec -it postgres -- bash

Run the script inside the pod:

psql -U wso2carbon -d wso2am\_db -f /tmp/postgresql.sql

\*\* using same steps to run database scripts on wso2\_shared\_db

**5. Modify values.yaml for PostgreSQL Configuration and Hostname**

1- Update the values.yaml file to reflect the PostgreSQL configuration:

wso2:

deployment:

databases:

apim\_db:

type: postgresql

host: postgres

port: 5432

username: wso2carbon

password: wso2carbon

database: wso2am\_db

shared\_db:

type: postgresql

host: postgres

port: 5432

username: wso2carbon

password: wso2carbon

database: wso2am\_shared\_db

### **2- Also modify in values.yaml**

* Updated gateway hostname to gw.localhost.
* Updated WebSocket hostname to websocket.localhost.
* Updated WebSub hostname to websub.localhost

am:

ingress:

gateway:

hostname: gw.localhost

websocket:

hostname: websocket.localhost

websub:

hostname: websub.localhost

**6. Add Database Driver to Minikube**

Copy the PostgreSQL JDBC driver to the WSO2 deployment:

1. Download the driver locally.
2. Copy the driver to the WSO2 pod:

kubectl cp postgresql-<version>.jar wso2-apim:/opt/wso2am/repository/components/lib/

**7. Update Deployment Configuration**

Modify the deployment.yaml to include the JDBC driver path:

volumeMounts:

- name: jdbc-driver

mountPath: /opt/wso2am/repository/components/lib

volumes:

- name: jdbc-driver

hostPath:

path: /path/to/postgresql-driver

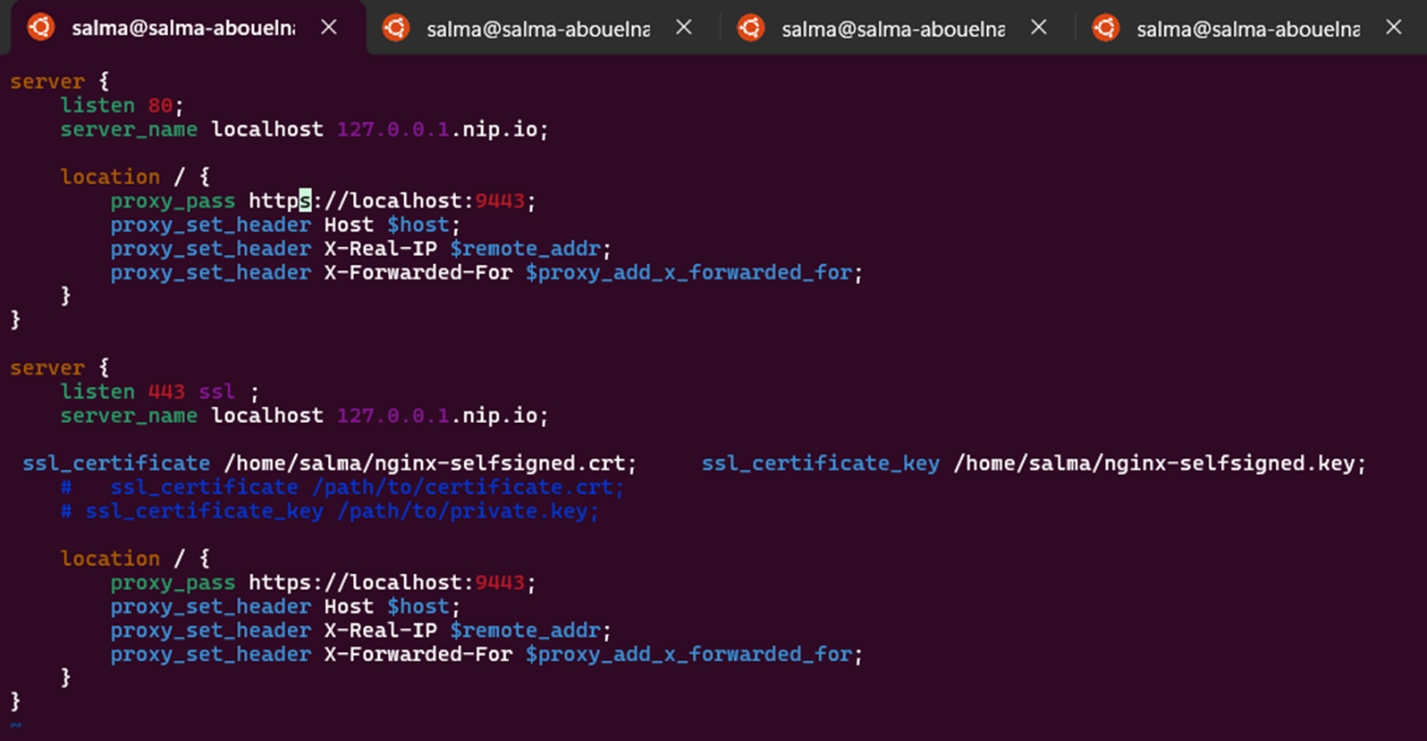
**8. Deploy WSO2 API Manager with Helm**

Run the following command to install the WSO2 API Manager:

helm install .

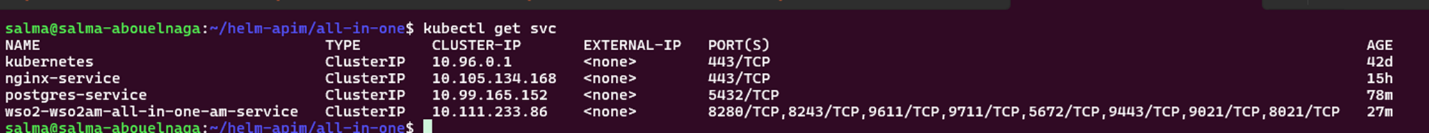
**9- Screenshots**

**1. Install and Configure Nginx**



**2. Install PostgreSQL as a Pod and Expose PostgreSQL as a service**:

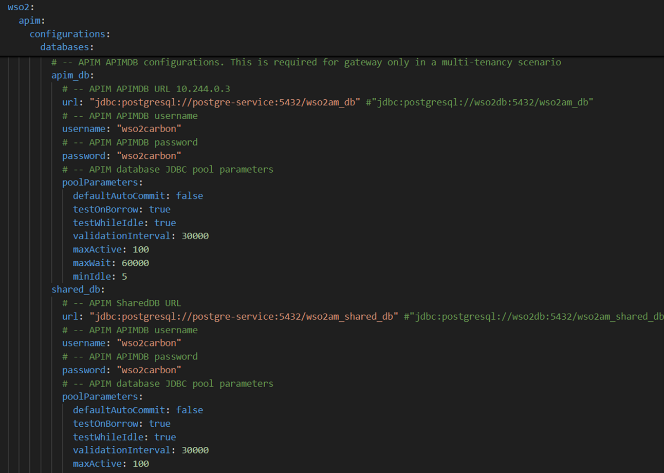


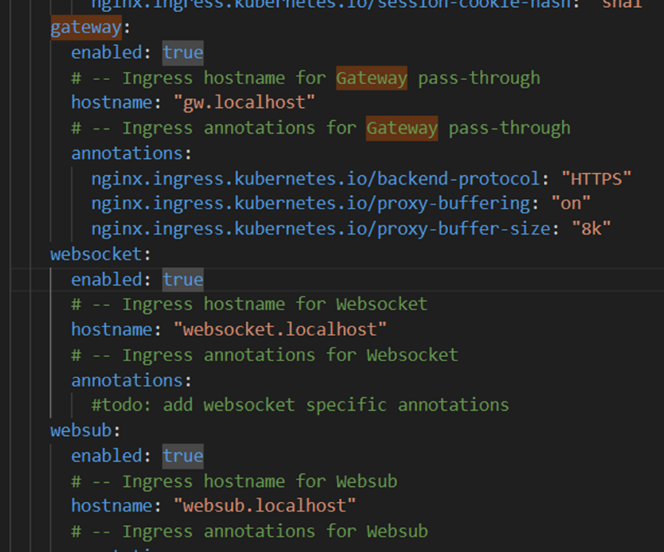


**3. Prepare and Run Database Scripts**



**4. Modify values.yaml for PostgreSQL Configuration and Hostname**

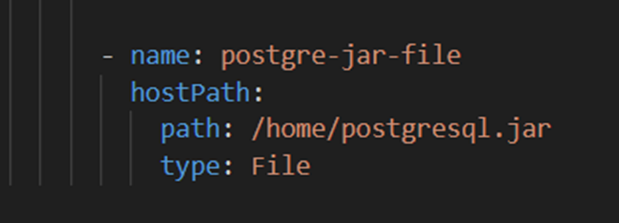




**5. Add Database Driver to Minikube**



**7. Update Deployment Configuration to include the JDBC driver path**



**Troubleshooting**

**Common Errors**

1. **502 Bad Gateway from Nginx**:
   * Verify Nginx points to the correct WSO2 service.
   * Check SSL certificates.
2. **Database Connection Issues**:
   * Verify PostgreSQL service is running.
   * Ensure correct credentials are in values.yaml.