Demo Project: Deploy Mosquitto message broker with ConfigMap and Secret Volume Types

This guide demonstrates how to deploy the Mosquitto message broker using Kubernetes with ConfigMap and Secret volume types to override its default configurations.

Step 1: Deploy Mosquitto Without Volumes

1. **Create a Mosquitto Deployment Without Volumes**:
    - Create the `mosquitto-without-volumes.yaml` file.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mosquitto
  labels:
    app: mosquitto
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mosquitto
  template:
    metadata:
      labels:
        app: mosquitto
    spec:
      containers:
        - name: mosquitto
          image: eclipse-mosquitto:2.0
          ports:
            - containerPort: 1883
```

- Apply the deployment: `kubectl apply -f mosquitto-without-volumes.yaml`

2. **Verify Pod Creation**:
   - Confirm the pod is running and retrieve its name: `kubectl get pod`
3. **Access the Pod Terminal**:
   - Open a shell into the Mosquitto pod: `kubectl exec -it <pod name> -- /bin/sh`
4. **Check the Default Configuration**:
   - View the default `mosquitto.conf` file: `cat /mosquitto/config/mosquitto.conf`

     Note: The default `mosquitto.conf` file will have everything commented out.
5. **Clean Up**:
   - Exit the pod terminal and delete the deployment:
     - `exit`
     - `kubectl delete -f mosquitto-without-volumes.yaml`

Step 2: Override Mosquitto Configuration with ConfigMap and Secret

Part 1: Create ConfigMap and Secret

1. **Prepare Configuration Files**:
    - Create a `config-file.yaml` for the ConfigMap:

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
    name: mosquitto-config-file
data:
    mosquitto.conf: |
        log_dest stdout
        log_type all
        log_timestamp true
        listener 9001
```

    - Create a `secret-file.yaml` for the Secret:

```yaml
apiVersion: v1
kind: Secret
metadata:
    name: mosquitto-secret-file
type: Opaque
data:
    secret.file: |
        VGVjaFdvcmxkMjAyMyEgLW4K
```

2. **Apply the ConfigMap and Secret**:
    - Deploy both files to the Kubernetes cluster:
        - `kubectl apply -f config-file.yaml`

- `kubectl apply -f secret-file.yaml`

3. **Verify Creation**:
   - Confirm that the ConfigMap and Secret were created:
     - `kubectl get configmap`
     - `kubectl get secret`

Part 2: Create Deployment With Volumes

1. **Create `mosquitto.yaml` Deployment File**:
   - Update the deployment to include volume mounts for the ConfigMap and Secret:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mosquitto
  labels:
    app: mosquitto
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mosquitto
  template:
    metadata:
      labels:
        app: mosquitto
    spec:
      containers:
        - name: mosquitto
          image: eclipse-mosquitto:2.0
          ports:
            - containerPort: 1883
          volumeMounts:
            - name: mosquitto-config
              mountPath: /mosquitto/config
            - name: mosquitto-secret
              mountPath: /mosquitto/secret
              readOnly: true
      volumes:
        - name: mosquitto-config
          configMap:
            name: mosquitto-config-file
        - name: mosquitto-secret
          secret:
            secretName: mosquitto-secret-file
```

2. **Apply the Deployment**:
   - Deploy the updated configuration: `kubectl apply -f mosquitto.yaml`
3. **Verify Deployment**:
   - Confirm the pod is running and retrieve its name:

     `kubectl get pod`
4. **Access the Pod Terminal**:
   - Open a shell into the Mosquitto pod:

     `kubectl exec -it <pod name> -- /bin/sh`

5. **Verify Configuration and Secret Contents**:
    - Check the overridden `mosquitto.conf` file:
      `cat /mosquitto/config/mosquitto.conf`
    - Check the Secret file:
      `cat /mosquitto/secret/secret.file`
