RabbitMQ Consumer HPA based on Queue depth:

Steps:

- Prerequisite:
  - kubectl get apiservices | grep "autoscaling"
    (it should show: v2beta2.autoscaling in output)

- Git Clone:
  - git clone
    https://github.com/sky29/rabbitmq-consumer-hpa-based-on-queue-depth.git

- RabbitMQ Broker:
  - cd 01-rabbitmq-broker/
  - kubectl apply -f .
  - Verify:
    - Command Line:
      - watch kubectl get pods
      - watch kubectl get svc
    - In browser:
      - <External_IP/DNS>:15672
      - <External_IP/DNS>:15692/metrics

- RabbitMQ Clients (Publisher & Consumer):
  - cd 02-rabbitmq-clients/
  - kubectl apply -f .
  - Verify:
    - Command Line:
      - watch kubectl get pods
      - watch kubectl get svc
    - In browser:
      - <External_IP/DNS>:9001
      - <External_IP/DNS>:9002
      - RabbitMQ management console queues tab

  - Send Messages through Publisher to Queue (which will be consumed by Consumer):
    - In browser:
      - <External_IP/DNS>:9001/send (ignore the blank screen on browser ……. thats fine since its just POC)
    - Verify in RabbitMQ management console queues tab

- Prometheus:
    - cd 03-prometheus/
    - kubectl apply -f .
    - Important Node: This is just a POC level prometheus installation, not production level.
    - Verify:
        - Command Line:
            - watch kubectl get pods
            - watch kubectl get svc
        - In browser:
            - <External_IP/DNS>:8080
            - Search in console:
                - rabbitmq_queue_messages{kubernetes_name="rabbitmq-management-service",kubernetes_namespace="default",queue="helloworld_queue"}
            - Also cross verify in graph

- Prometheus Adapter:
    - cd 04-prometheus-adapter/

    - cd certs/
    - ./gencerts.sh

    - kubectl create ns custom-metrics
    - kubectl apply -f cm-adapter-serving-certs.yaml -n custom-metrics

    - cd manifests/
    - vi custom-metrics-apiserver-deployment.yaml
      update: [--prometheus-url=http://prometheus-service.default.svc.cluster.local:8080/] as per your environment
    - vi custom-metrics-config-map.yaml:
      update: [queue="helloworld_queue"] queuename as per your environment
    - kubectl apply -f .

    - Verify:
        - Command Line:
            - kubectl get --raw /apis/custom.metrics.k8s.io/v1beta1
            - kubectl get --raw /apis/custom.metrics.k8s.io/v1beta1/namespaces/default/services/rabbitmq-management-service/rabbitmq_queue_messages?metricLabelSelector=queue%3Dhelloworld_queue

                - Send messages through publisher again & verify above url

- HPA:
    - cd 05-hpa/
    - kubectl apply -f hpa.yaml

    - Verify:
        - Command Line:
            - watch kubectl get pods
            - Send messages through publisher again & watch pod autoscaling

    - Note:
        - Client Program is currently not using "prefetch count", that's way message distribution through consumer might not be visible in efficient manner