Workshop: Zero data loss messaging for high available workloads with Red Hat AMQ Broker

# Objective

This workshop will guide participants through the process of configuring and testing a mirrored setup in a GitOps-enabled OpenShift cluster. By following these steps, participants will verify system failover mechanisms and ensure high availability of applications.

# Prerequisites

* Access to an OpenShift cluster with GitOps configured
* Knowledge of OpenShift UI navigation
* Mirrored broker setup is deployed with corresponding client and publisher applications

# Steps to Follow

## Step 1: Verify the Mirrored Setup

1. Open the **OpenShift Web Console** and go to **Workloads > Pods**
2. Ensure the setup of the broker pods are in the mirrored configuration
   1. The mirrored broker configuration contains two statefulsets, while the cluster configuration contains two pods of the same statefulset.
3. If not, ensure the ArgoCD application is pointed to the mirrored branch.

### Step 1a: Restart the Secondary Broker

**Using UI:**

1. In the OpenShift Web Console, go to **Workloads > Pods**.
2. Locate the secondary broker pod (dc2) and delete it to force a restart.
3. Monitor the pod status to confirm a successful reconnection.

This step is added to ensure the connections go via the broker on the node we will be restarting

**Using CLI:**

*oc delete pod <secondary-broker-pod-name> -n <namespace>*

## Step 2: Start the Producer Application

1. Open a web browser and navigate to the **URL exposed via the publisher route**.
   1. The URL can be found in **Networking > Routes**
2. Initiate data publishing by interacting with the UI.

## Step 3: Confirm Publisher and Client Are Working

**Using UI:**

1. Navigate to **Workloads > Pods**.
2. Click on the **Publisher** and **Client** pods.
3. View the logs under the **Logs** tab.

**Using CLI:**

*oc logs -f <publisher/client-pod-name> -n <namespace>*

## Step 4: Identify the Primary Broker Node

**Using UI:**

1. Go to **Workloads > Pods**.
2. Locate the **Primary Broker Pod** and click on it.
3. Under the **Details** tab, find the **Node** information.

**Using CLI:**

*oc get pods -o wide -n <namespace>*

## Step 5: Initiate a node failure

1. Due to permissions in Azure, only the instructors have access to the Azure VM corresponding to the node.
2. Please ask the instructors to stop the **Node** corresponding to the primary broker.

## Step 6: Observe the Failover in OpenShift

**Using UI:**

1. Go to **Workloads > Pods**.
2. Monitor the **Client** and **Publisher** pod logs to observe failover behavior.

**Using CLI:**

*oc logs -f <publisher/client-pod-name> -n <namespace>*

1. The system should detect the failure and switch to the secondary broker automatically.
2. Verify that message flow resumes via the new broker instance.

## Step 7: Cleanup

1. Please ask the instructors to restart the Azure VM.
2. Allow time for the VM to become fully operational.
3. Ensure that it rejoins the OpenShift cluster and resumes its role as the primary broker **Node**.