E-Commerce Customer Order Processing With SQS

Consider an eCommerce Portal that makes several products available for purchase online by customers. A customer can make a purchase by placing an order, and when the order is confirmed, an email notification is also sent to the customer, along with the product being scheduled for delivery.

The eCommerce Portal has an Order Processing Subsystem that takes care of the above scenario. This subsystem gets real-time order details from various customers, and places them into an SQS queue. The purpose of this SQS queue is mainly to facilitate later processing, where email notifications are sent to the registered email IDs of these customers, through the SNS service.

You can assume each Order to have the following details:

{
'customer_name': <customer name - STRING>,
'email_id': <email ID - STRING>,
'product _id': <product ID - STRING>,
'product_name': <product name - STRING>,
'price': <product price - FLOAT>,
'timestamp': <timestamp of purchase - LONG INTEGER or DATE, whichever is convenient>
}

Write a Python script that generates a large number of such order records, and enqueues them one-by-one into the SQS queue. Such a Python script could run on an EC2 instance that has

access to the SQS queue. Once an order record is enqueued into the SQS queue, it is available for further processing.

Write a background Python application that takes records one-by-one from the SQS queue, and generates an SNS notification for the customer associated with that order. Such an application could run on another EC2 instance that has access to the same SQS queue.

Note that the frequency with which orders are enqueued into the SQS queue, need not match with the frequency with which they are dequeued and processed. You can deliberately ensure that these frequencies of queue access are different.

This is actually an extremely common use case of SQS, since an eCommerce Portal may not want to send notification mails in the same API call as the original order - since that would make the customer wait. Mails can always go later, in a matter of a few minutes.