

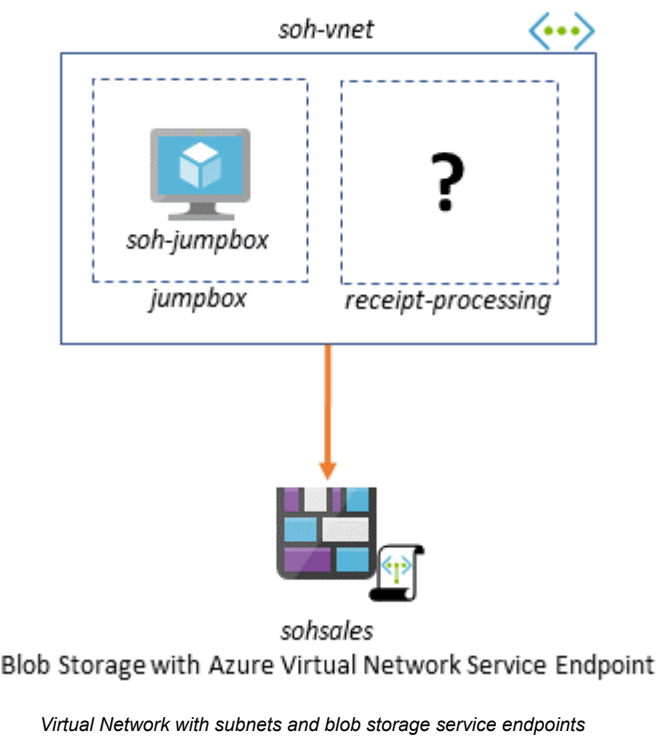
# Pub/Sub and Network Integration

## Background

Executives at Best For You Organics Company (BFYOC) have taken notice of the success of the recently implemented solution to enable Point of Sale (POS) terminals to send sales data to the cloud, and they want to leverage this new data even more. They have started a pilot program to update BFYOC POS terminals to randomly capture a PDF version of the sales receipt in addition to the detailed transaction data. They would like to save all the transaction data to their private storage in the cloud for data archival and future analysis. Additionally, executives are concerned about the potential for fraudulent transactions, and would like to add additional rigor for transactions which exceed \$100.

After learning of the executive's high-level requirements, the cloud architecture team has decided to implement a publisher/subscriber (pub/sub) messaging technique to capture POS sales data which has a receipt. They feel they can use a message filtering pattern to identify sales data which has a total transaction cost greater than \$100. Additionally, because BFYOC is highly protective of customer and sales data, they want to ensure all the data is safely stored within an Azure storage account which is accessible only from services within, or integrated with, BFYOC's virtual network in Azure. The files stored in blob storage are able to be used by other systems within BFYOC's virtual network, such as data analysis and fraud detection processes.

BFYOC has an existing virtual network containing at least two subnets. They also have an Azure storage account configured to use Azure Virtual Network service endpoints to restrict access from any source except the two subnets within the virtual network. An Azure Virtual Machine is available to be used as a jumpbox for viewing data from services within the virtual network.



## Challenge

Your first challenge is to modify the Azure Function used in the previous challenge to send the specified receipt information to a pub/sub messaging system, but only if there is a receipt. Using the POS data sent via the Event Hub from the previous challenge, you can create a new JSON document with a structure similar to the example below that will be sent to your pub/sub messaging system.

```
{
  "totalItems": 8,
  "totalCost": 123.40,
  "salesNumber": "0c423398-3c7c-0682-7519-4701c445ed7a",
  "salesDate": "09/11/2019 06:04:43",
  "storeLocation": "00d8ea6f-935c-2cca-9bbc-f56b5a091621",
  "receiptUrl": "https://serverlessohsales.blob.core.windows.net/TheReceipt.pdf"
}
```

Next, you will need to create a process(es) which is able to filter based on the total cost in each message, and accomplish the following tasks:

- 1. If the total cost is greater than or equal to \$100:
  - o retrieve the PDF file of the receipt using the receiptUrl value
  - o base64 encode the PDF
  - o create a JSON object which includes the receipt data, along with the base64 encoded version of the receipt
  - o save the JSON object with a unique name to the “receipts-high-value” container within the provided Azure Storage account (accessible only from the virtual network)

The format for the saved data should be as follows:

```
{
  "Store": "00d8ea6f-935c-2cca-9bbc-f56b5a091621",
  "SalesNumber": "0c423398-3c7c-0682-7519-4701c445ed7a",
  "TotalCost": 123.40,
  "Items": 8,
  "SalesDate": "09/11/2019 06:04:43",
  "ReceiptImage": "V2VsY29tZSB0byBTZXJ2ZXJsZXNzIE9wZW5lYWNRlQ=="
}
```

- 2. If the total cost is less than \$100:
  - o create a JSON object which includes the receipt data
  - o save the JSON object with a unique name to the “receipts” container within the provided Azure Storage account (accessible only from the provided virtual network)

The format for the saved data should be as follows:

```
{
  "Store": "00d8ea6f-935c-2cca-9bbc-f56b5a091621",
  "SalesNumber": "0c423398-3c7c-0682-7519-4701c445ed7a",
  "TotalCost": 6.58,
  "Items": 1,
  "SalesDate": "09/02/2019 10:36:17"
}
```

When determining how to perform the filtering of messages, it is important to remember that BFYOC has built a distributed system which may, over time, require additional processing streams. The cloud architecture team wants to ensure the solution avoids filtering “in code” by leveraging features of the selected messaging system.

HINT: All receipts should be sent to the messaging system, regardless of overall total cost.

## Success Criteria

- Demonstrate the Azure Function which publishes data to a pub/sub messaging service.
- Demonstrate the ability to filter messages to the appropriate process based on the total cost of purchased items.
- Demonstrate the ability to retrieve messages from the pub/sub service and save JSON files with a unique name to the proper blob storage container in the provided Azure Storage account (which uses service endpoints).

## References

- [Claim-Check Pattern \(https://docs.microsoft.com/azure/architecture/patterns/claim-check\)](https://docs.microsoft.com/azure/architecture/patterns/claim-check)
- [Azure Virtual Network Service Endpoints \(https://docs.microsoft.com/azure/virtual-network/virtual-network-service-endpoints-overview\)](https://docs.microsoft.com/azure/virtual-network/virtual-network-service-endpoints-overview)
- [Azure Functions Premium plan \(network connectivity\) \(https://docs.microsoft.com/azure/azure-functions/functions-premium-plan#private-network-connectivity\)](https://docs.microsoft.com/azure/azure-functions/functions-premium-plan#private-network-connectivity)
- [Azure Functions binding expression patterns \(https://docs.microsoft.com/azure/azure-functions/functions-bindings-expressions-patterns#create-guids\)](https://docs.microsoft.com/azure/azure-functions/functions-bindings-expressions-patterns#create-guids)
- [Azure Functions triggers and bindings concepts \(https://docs.microsoft.com/azure/azure-functions/functions-triggers-bindings\)](https://docs.microsoft.com/azure/azure-functions/functions-triggers-bindings)
- [Azure Functions Recipes - Triggers and bindings \(https://docs.microsoft.com/sandbox/functions-recipes/triggers-bindings\)](https://docs.microsoft.com/sandbox/functions-recipes/triggers-bindings)

- ## Progress Diagram

