# E-Commerce Order History App (without Lambda)

## Set up fake order log generator:

* Set up EC2 T2/micro instance with latest Amazon Linux AMI
* **sudo yum install –y aws-kinesis-agent**
* wget <http://media.sundog-soft.com/AWSBigData/LogGenerator.zip>
* unzip LogGenerator.zip
* chmod a+x LogGenerator.py
* sudo mkdir /var/log/cadabra

## Set up Kinesis Streams:

* AWS console: Kinesis / create data stream
  + Warn student that Kinesis streams have an hourly cost whether you're using them or not
* Name: CadabraOrders / 1 shard
* cd /etc/aws-kinesis
* sudo nano agent.json

{

"cloudwatch.emitMetrics": true,

"kinesis.endpoint": "",

"firehose.endpoint": "",

"awsAccessKeyId": "AKIAJF4KVZWOY65QWOOA",

"awsSecretAccessKey": "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",

"flows": [

{

"filePattern": "/var/log/cadabra/\*.log",

"kinesisStream": "CadabraOrders",

"partitionKeyOption": "RANDOM",

"dataProcessingOptions": [

{

"optionName": "CSVTOJSON",

"customFieldNames": ["InvoiceNo", "StockCode", "Description", "Quantity", "InvoiceDate", "UnitPrice", "Customer", "Country"]

}

]

}

]

}

* sudo service aws-kinesis-agent start
* **sudo chkconfig aws-kinesis-agent on**

## Test the stream:

* **cd ~**
* **sudo ./LogGenerator.py**
* **tail -f /var/log/aws-kinesis-agent/aws-kinesis-agent.log**

## Set up DynamoDB:

* Create CadabraOrders table
* Primary key: CustomerID
* Sort key: OrderID
* Set provisioning to stay within free tier

## Set up consumer app:

* sudo pip install boto3
* Create ~/.aws/credentials:
* [default]  
  aws\_access\_key\_id = YOUR\_ACCESS\_KEY  
  aws\_secret\_access\_key = YOUR\_SECRET\_KEY
* Create ~/.aws/config:
* [default]  
  region=us-east-1
* wget http://media.sundog-soft.com/AWSBigData/Consumer.py
* Chmod a+x Consumer.py
* ./Consumer.py
* In another window: sudo ./LogGenerator.py 10
* Give it a minute, observe Consumer script processing the 10 new lines
* Observe Items in the DynamoDB table in the console