AWS Lab environment

* Start NuvePro Lab
* Check region
* N.Virginia (us-east-1)

Local system environment

* Create following folder in the directory 'C:'
* AWSCloud-Lab-02
* Create a text file ‘Lab-02-Info.txt’ within it
* Create a folder ‘Screenshots’ within it

Lab-02-Info.txt template

# Source Bucket Name

-

# Target Bucket Name -

-

# Topic ARN

-

# EC2's Public IP Address -

# MySQL DB Username - admin

# MySQL DB Password - password

# MySQL DB Name - invoice\_db

# MySQL DB Endpoint -

S3 Buckets creation

* Source Bucket
* doc-processor-source-invoice-bucket-nov-2024-<username>
* Target Bucket
* doc-processor-target-invoice-bucket-may-2024-<username>
* Screenshot
* S3 buckets in dashboard

Topic creation

* Type
* Standard
* Name
* Invoice\_notification\_topic
* Screenshot
* topic details page

Access policy for Topic

* <https://docs.aws.amazon.com/AmazonS3/latest/userguide/grant-destinations-permissions-to-s3.html#grant-sns-sqs-permission-for-s3>
* {
* "Version": "2012-10-17",
* "Id": "example-ID",
* "Statement": [
* {
* "Sid": "Example SNS topic policy",
* "Effect": "Allow",
* "Principal": {
* "Service": "s3.amazonaws.com"
* },
* "Action": [
* "SNS:Publish"
* ],
* "Resource": "**SNS-topic-ARN**",
* "Condition": {
* "ArnLike": {
* "aws:SourceArn": "arn:aws:s3:\*:\*:**amzn-s3-demo-bucket**"
* },
* "StringEquals": {
* "aws:SourceAccount": "**bucket-owner-account-id**"
* }
* }
* }
* ]
* }
* Screenshot
* Access policy editor page

Event notification rule on S3 bucket

* On the source S3 bucket
* Properties -> Event Notification
* Event Name
* InvoiceAddedEvent
* Event Type
* Put

s3:ObjectCreated: Put

* Destination
* SNS Topic
* Specify SNS Topic
* Invoice\_notification\_topic
* Screenshot
* Event Notifications section

Security Group for EC2

* Security Group Name
* python\_server\_ssh\_http\_access
* Description
* To access Python Server / Application via SSH and HTTP Protocols
* Inbound Rules
* Rule 1
* Type - SSH
* Port - 22
* Source - Anywhere IPv4
* Rule 2
* Type - Custom TCP
* Port 8080
* Source - Anywhere IPv4
* Outbound Rules
* Rule 1
* Type - All Traffic
* Protocol - All
* Destination - Anywhere - IPv4

EC2 instance launch

* Name
* Python-Server-Invoice-Processor
* AMI
* Amazon Linux
* Instance Type
* t2.micro
* Create new Key Pair
* Key Pair Name
* invoice-processor-keypair
* Key Pair Type
* RSA
* Private key File Format
* .pem
* Network Settings
* Firewall Settings
* Select Existing Security Group
* python\_server\_ssh\_http\_access
* Screenshot
* EC2 details in dashboard

IAM Role

* Trusted Entity Type
* Choose 'AWS Service'
* Use Case
* Service or Use-case
* Choose 'EC2'
* Use-Case
* Ensure that 'EC2' is selected
* Policies
* AmazonS3FullAccess
* AmazonRDSFullAccess
* Role name
* Python-Server-App-RDS-S3-Access

Modify IAM role of EC2

* Security -> Modify IAM Role
* Screenshot
* IAM role update page with the selected role in dropdown

Security group for RDS

* Security Group Name
* mysql\_access
* Description
* To access MySQL RDS database instance
* Inbound Rules
* Rule 1
* Type - MySQL / Aurora
* Port - 3306
* Source - Anywhere IPv4
* Outbound Rules
* Rule 1
* Type - All Traffic
* Protocol - All
* Destination - Anywhere - IPv4

RDS Creation

* Create Database
* Standard create
* Engine
* Mysql
* Engine Version
* Default
* Templates
* Dev/Test
* Availability & Durability
* Single DB instance
* Settings
* DB instance identifier
* python-app-rds-instance
* Credentials settings
* Master username
* admin
* In Credentials management section, Choose 'Self managed
* Master password
* password
* Instance configuration
* DB instance class
* Burstable classes
* db.t3.micro
* Storage
* Storage type
* General Purpose SSD (gp2)
* Connectivity
* Don’t connect to an EC2 compute resource
* Public access
* **Yes**
* (This is very important. If it is not set to Yes, the database will not be available for access from anywhere)
* VPC security group
* Remove the 'default' security group selected
* Choose ‘mysql\_access’ sec group
* Availability Zone
* No Preference
* Additional configuration
* Database Port
* 3306
* Database authentication
* Password authentication
* Additional Configuration
* Database options
* Initial Database name
* invoice\_db
* Screenshot
* RDS database details page

Using MySQL WorkBench to Verify check connection to RDS

Once installed, create profile and test connection to RDS

* Connection Name
* Python-RDS-App
* Connection Method
* Default (Standard TCP/IP)
* Hostname
* <RDS Endpoint>
* User
* admin
* Password
* password
* In New Query tab
* use invoice\_db;

Download Source Code

On CloudShell, run the following commands

cd ~

git clone <https://github.com/joycee-gl/Invoice-Processor.git>

ls

sudo vi ./Invoice-Processor/api/vews.py

* Update the hostname and s3\_target\_bucket in views.py
* Screenshot
* with the changes done in views.py

Configuring and setting up the application

chmod 400 ./invoice-processor-keypair.pem

PYTHON\_SERVER\_IP=54.211.2.214

PEM\_FILE=./invoice-processor-keypair.pem

scp -i $PEM\_FILE -r ./Invoice-Processor ec2-user@$PYTHON\_SERVER\_IP:~

Screenshot

* with the scp command result on console

-----------------------------------------

ssh -i $PEM\_FILE ec2-user@$PYTHON\_SERVER\_IP

cd ~

ls Invoice-Processor

Screenshot

* with the ls command result on console

Configure the Application

sudo cp -r Invoice-Processor /opt

sudo chown ec2-user:ec2-user -R /opt

cd /opt/Invoice-Processor

python3 --version

python3 -m ensurepip --upgrade

python3 -m pip install --upgrade pip

pip3 --version

pip3 install virtualenv

virtualenv --version

virtualenv inv\_proc

source inv\_proc/bin/activate

pip3 install django boto3 mysql-connector-python

python3 manage.py migrate

-----------------------------------------

Starting the Application

python3 manage.py runserver 0:8080

-----------------------------------------

Verification of Application

In a new cloudshell tab,

PYTHON\_SERVER\_IP=54.211.2.214

PEM\_FILE=./invoice-processor-keypair.pem

ssh -i $PEM\_FILE ec2-user@$PYTHON\_SERVER\_IP

curl [http://localhost:8080](http://localhost:8080/)

-----------------------------------------

Configure Http Endpoint as Subscription in Topic

* Protocol
* Http
* Endpoint
* [http://<EC2-Public-IP>:8080/sns](http://%3Cec2-public-ip%3E:8080/sns)
* Screenshots
* cloudshell console with SubscribeURL displayed
* subscription confirmation (XML response) in the browser
* subscription confirmation in the Subscriptions section in Topic

--------------------------------------------

Trigger the end-to-end scenario

cd ~

aws s3 cp ./Invoice-Processor/sample-invoice/docproc-invoice.txt s3://doc-processor-source-invoice-bucket-nov-2024-jc/docproc-invoice.txt

Verify Target bucket contents

* Check the target bucket for the presence of generated CSV
* Screenshot
* page with CSV in the target bucket

Verify DB contents

use invoice\_db;

select \* from invoice;

--------------------------------------