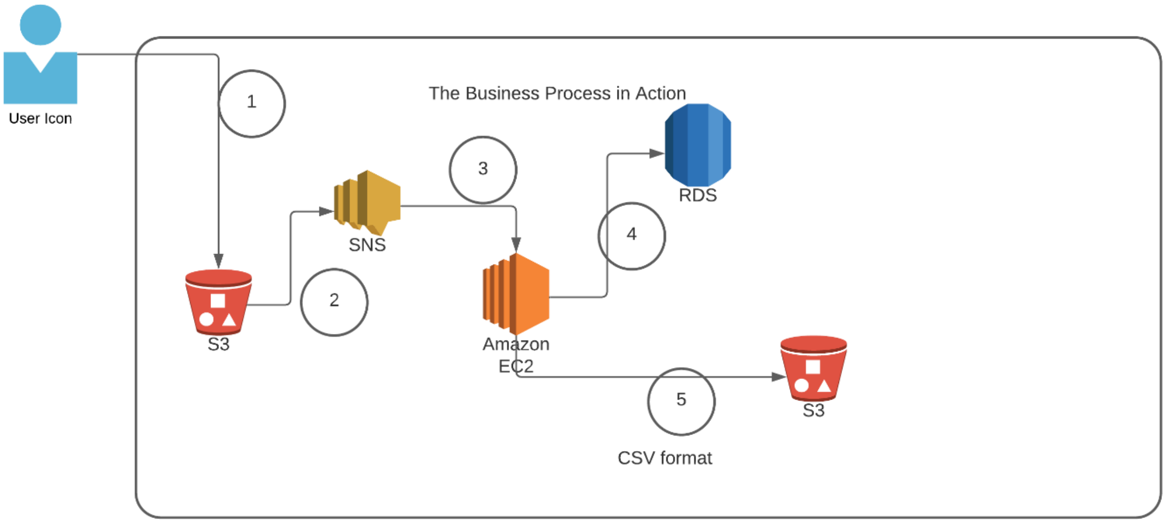
Dr. Shubhangi Kharche

AWS Project 2

Architecture diagram

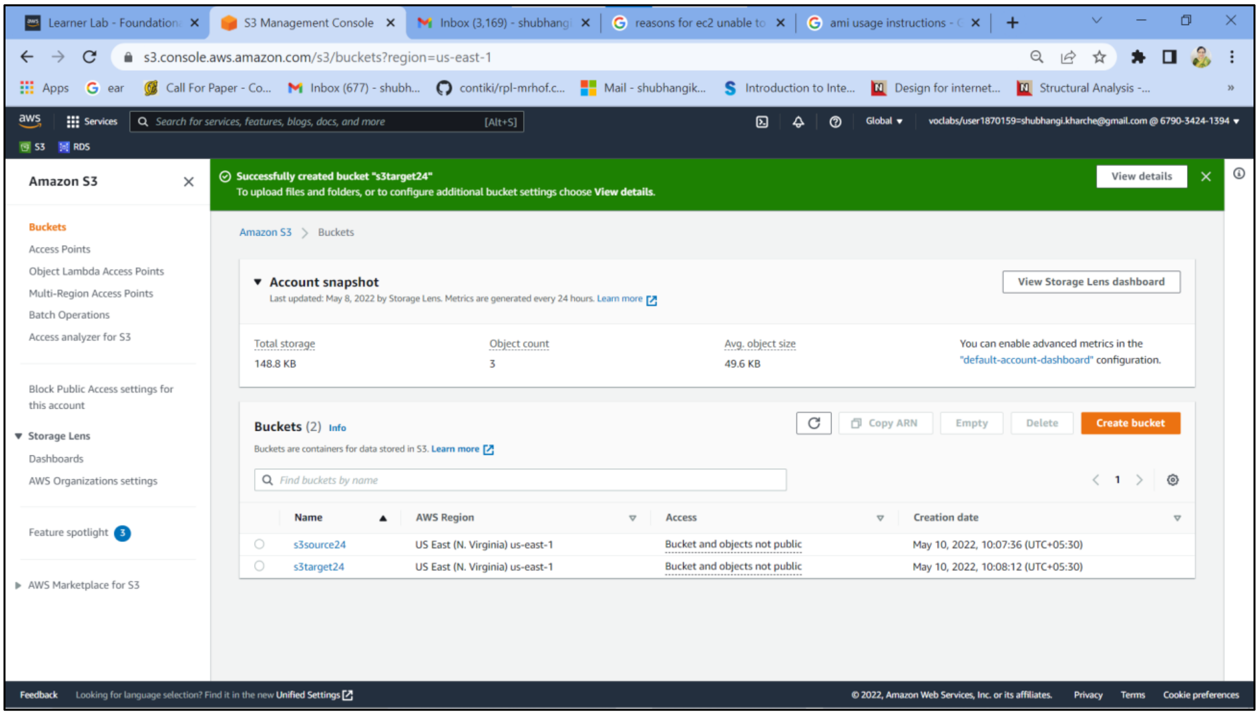


Architecture Implementation	
1	The customer uploads the invoice data to S3 bucket in a text format as per their guidelines and policies. This bucket will have a policy to auto delete any content that is more than 1 day old (24 hours).
2	An event will trigger in the bucket that will place a message in SNS topic
3	A custom program running in EC2 will subscribe to the SNS topic and get the message placed by S3 event
4	The program will use S3 API to read from the bucket, parse the content of the file and create a CSV record and save the details in an RDS database
5	The program will use S3 API to write CSV record to destination S3 bucket as new S3 object.

Skills: Amazon S3, Amazon EC2, Amazon RDS, Amazon SNS, API, Python, Boto3

Step 1: SNS and S3 topic creation

Step number	a
Step name	Creation of Source and target buckets
Instructions	1) Navigate to S3 using the Services button at the top of the screen 2) Select "Create Bucket" 3) Enter a source bucket name and use the default options for the rest of the fields 4) Click on "Create Bucket' 5) Repeat the above steps to create a target bucket
Expected screenshots	1) Screen showing created S3 source and target buckets



Successfully created bucket "s3target24"
To upload files and folders, or to configure additional bucket settings choose **View details**.

Account snapshot
Last updated: May 8, 2022 by Storage Lens. Metrics are generated every 24 hours. Learn more

Total storage	Object count	Avg. object size	
148.8 KB	3	49.6 KB	You can enable advanced metrics in the "default-account-dashboard" configuration.

Buckets (2) Info
Buckets are containers for data stored in S3. Learn more

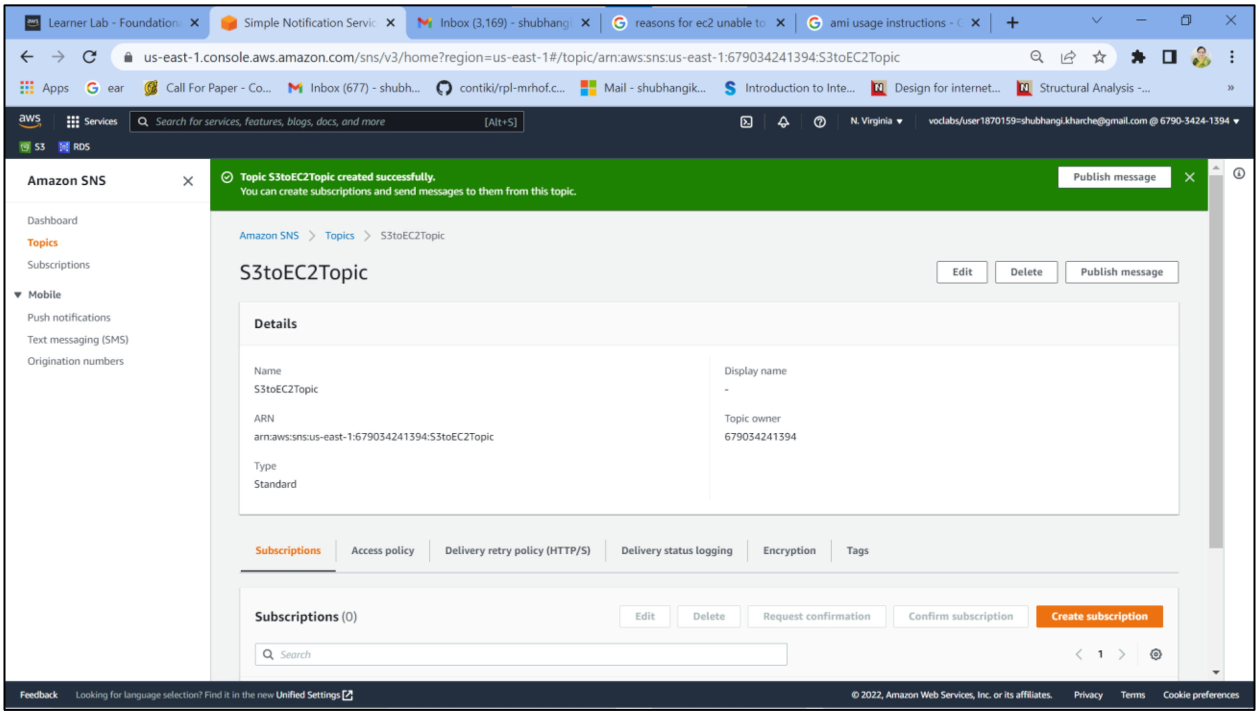
	Name	AWS Region	Access	Creation date
○	s3source24	US East (N. Virginia) us-east-1	Bucket and objects not public	May 10, 2022, 10:07:36 (UTC+05:30)
○	s3target24	US East (N. Virginia) us-east-1	Bucket and objects not public	May 10, 2022, 10:08:12 (UTC+05:30)

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates.

Fig1: Screen showing created S3 source and target buckets

Step number	b
Step name	Creation of SNS subscription
Instructions	1) Navigate to SNS -> Topics 2) Click on "Create Topic" 3) Enter the following fields Name : S3toEC2Topic The other options can be ignored for now 4) Click on Create Topic
Expected screenshots	1) Creation of SNS topic



Topic S3toEC2Topic created successfully.
You can create subscriptions and send messages to them from this topic.

S3toEC2Topic

Details

Name
S3toEC2Topic

ARN
arn:aws:sns:us-east-1:679034241394:S3toEC2Topic

Type
Standard

Display name
-

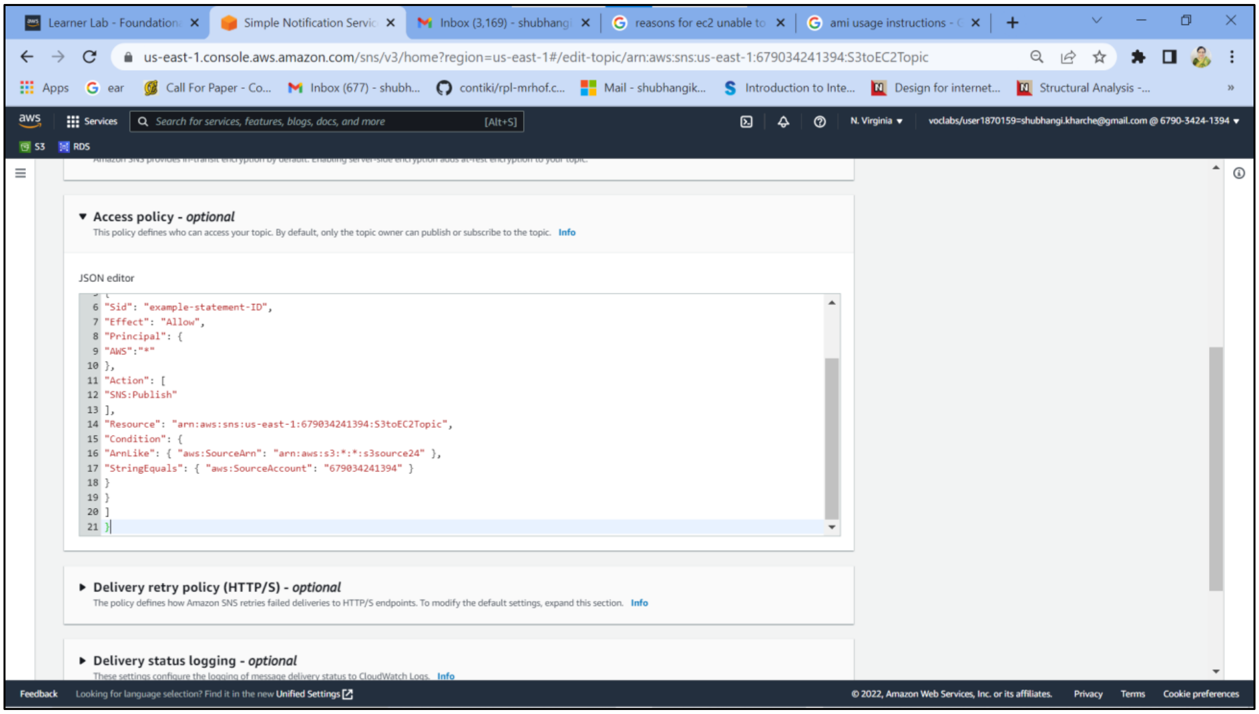
Topic owner
679034241394

Fig 2: Creation of SNS topic

Step number	c
Step name	Modification of SNS Access Policy

Instructions	1) Navigate to SNS -> Topics and select the topic created in the previous step
	2) Note down the ARN shown in the topic details
	2) Click on Edit and select "Access Policy".
	3) Replace the text in the JSON editor with the following
	{
	"Version": "2012-10-17",
	"Id": "example-ID",
	"Statement": [
	{
	"Sid": "example-statement-ID",
	"Effect": "Allow",
	"Principal": {
	"AWS":"*"
	},
	"Action": [
	"SNS:Publish"
	],
	"Resource": "**SNS-topic-ARN**",
	"Condition": {
	"ArnLike": { "aws:SourceArn": "arn:aws:s3:*:*:**bucket-name**" },
	"StringEquals": { "aws:SourceAccount": "**bucket-owner-account-id**" }
	}
	}
	]
	}
	
	4) Replace the bold text with the SNS topic ARN, source bucket name and your AWS account ID respectively.
	5) Click on Save Changes

Expected screenshots	1) JSON Editor screen



us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/edit-topic/arn:aws:sns:us-east-1:679034241394:S3toEC2Topic

Access policy - *optional*

This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. Info

JSON editor

```
{
"Sid": "example-statement-ID",
"Effect": "Allow",
"Principal": {
"AWS":"*"
},
"Action": [
"SNS:Publish"
],
"Resource": "arn:aws:sns:us-east-1:679034241394:S3toEC2Topic",
"Condition": {
"ArnLike": { "aws:SourceArn": "arn:aws:s3:*:*:s3source24" },
"StringEquals": { "aws:SourceAccount": "679034241394" }
}
}
]
}
```

Delivery retry policy (HTTP/S) - *optional*

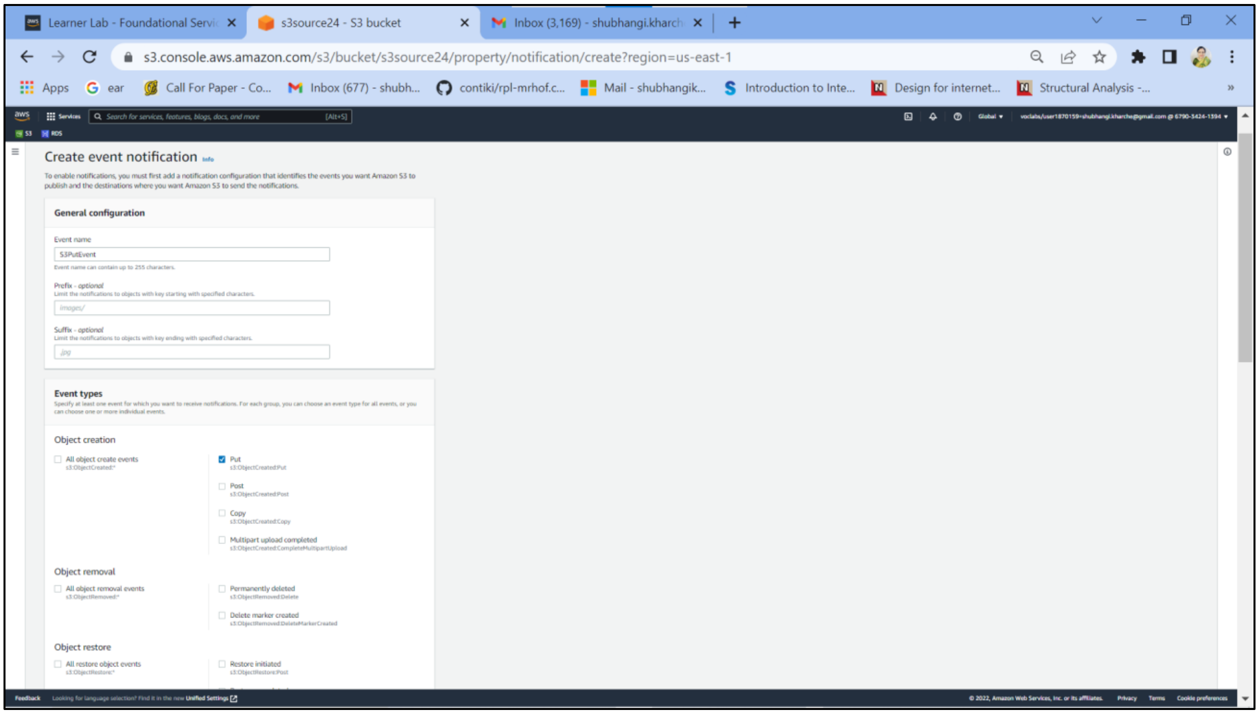
The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section. Info

Delivery status logging - *optional*

These settings configure the logging of message delivery status to CloudWatch Logs. Info

Fig 3: JSON Editor screen

Step number	d
Step name	Configuring SNS notifications for S3
Instructions	1) Navigate to S3 and select the source bucket created in Step 1 (a) 2) Select Properties and scroll down to Event Notifications and select it 3) Select "Create Event Notification" 4) Fillup the details as follows Name : S3PutEvent Select PUT from the list of radio buttons Destination : Select SNS Topic SNS : Select S3ToEC2Topic 5) Save Changes
Expected screenshots	1) Event Configuration Screen



s3.console.aws.amazon.com/s3/bucket/s3source24/property/notification/create?region=us-east-1

Create event notification

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

General configuration

Event name: S3PutEvent
Event name can contain up to 255 characters.

Prefix - *optional*
Limit the notifications to objects with key starting with specified characters.
images/

Suffix - *optional*
Limit the notifications to objects with key ending with specified characters.
.jpg

Event types

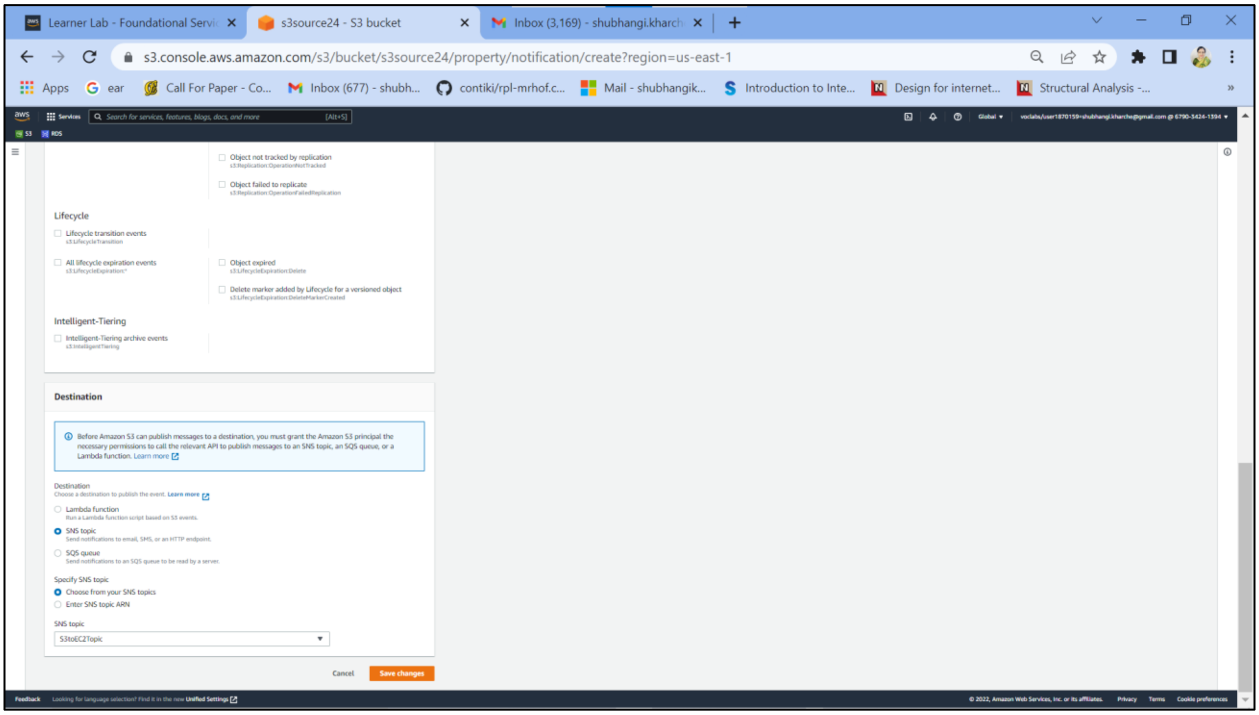
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation
- □ All object create events (s3:ObjectCreated:*)
- ☑ Put (s3:ObjectCreated:Put)
- □ Post (s3:ObjectCreated:Post)
- □ Copy (s3:ObjectCreated:Copy)
- □ Multipart upload completed (s3:ObjectCreated:CompleteMultipartUpload)

Object removal
- □ All object removal events (s3:ObjectRemoved:*)
- □ Permanently deleted (s3:ObjectRemoved:Delete)
- □ Delete marker created (s3:ObjectRemoved:DeleteMarkerCreated)

Object restore
- □ All restore object events (s3:ObjectRestore:*)
- □ Restore initiated (s3:ObjectRestore:Post)

Fig 4 (a): **Event Configuration Screen-1**



- □ Object not tracked by replication (s3:Replication:OperationNotTracked)
- □ Object failed to replicate (s3:Replication:OperationFailedReplication)

Lifecycle
- □ Lifecycle transition events (s3:LifecycleTransition)
- □ All lifecycle expiration events (s3:LifecycleExpiration:*)
- □ Object expired (s3:LifecycleExpiration:Delete)
- □ Delete marker added by Lifecycle for a versioned object (s3:LifecycleExpiration:DeleteMarkerCreated)

Intelligent-Tiering
- □ Intelligent-Tiering archive events (s3:IntelligentTiering)

Destination

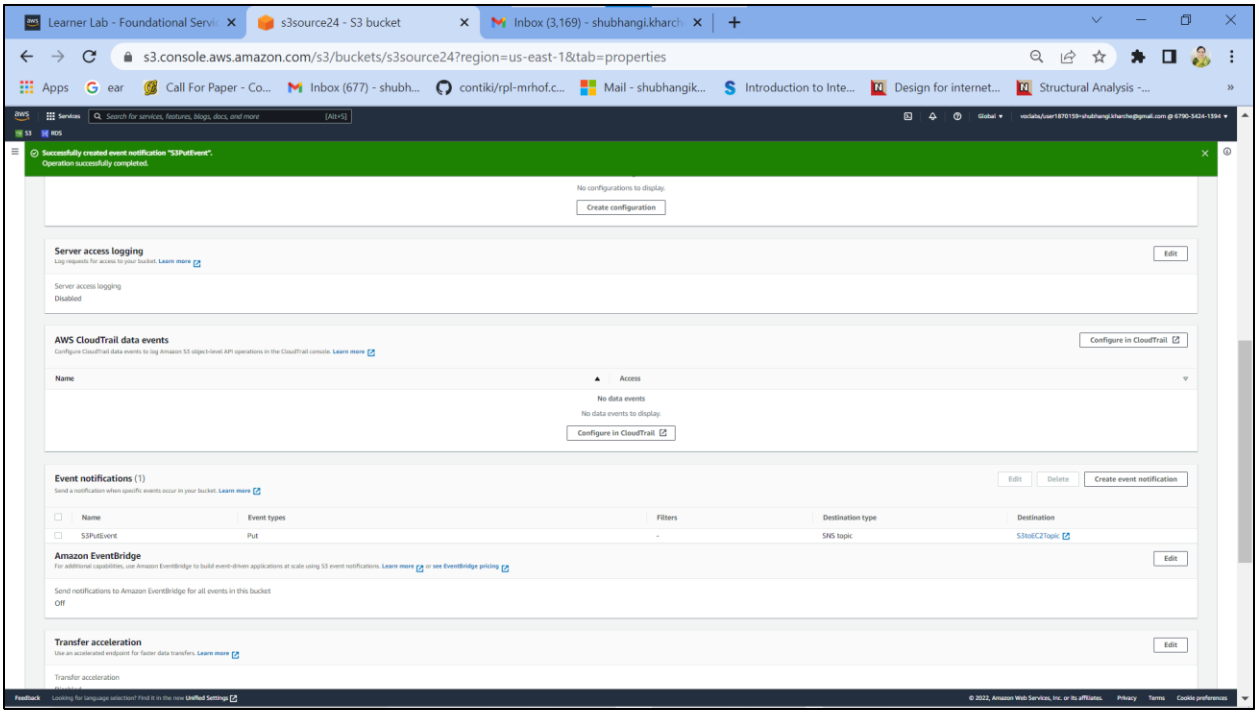
Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. Learn more

Destination
Choose a destination to publish the event. Learn more
- ○ Lambda function: Run a Lambda function script based on S3 events.
- ● SNS topic: Send notifications to email, SMS, or an HTTP endpoint.
- ○ SQS queue: Send notifications to an SQS queue to be read by a server.

Specify SNS topic
- ● Choose from your SNS topics
- ○ Enter SNS topic ARN

SNS topic: S3toEC2Topic

Fig 4 (b): **Event Configuration Screen-2**



Learner Lab - Foundational Servic
s3source24 - S3 bucket
Inbox (3,169) - shubhangi.kharch
s3.console.aws.amazon.com/s3/buckets/s3source24?region=us-east-1&tab=properties
Apps
ear
Call For Paper - Co...
Inbox (677) - shubh...
contiki/rpl-mrhof.c...
Mail - shubhangik...
Introduction to Inte...
Design for internet...
Structural Analysis -...
Services
Search for services, features, blogs, docs, and more
[Alt+S]
S3
RDS
Global
voclabs/user1870159=shubhangi.kharche@gmail.com @ 6790-3424-1394

Successfully created event notification "S3PutEvent".
Operation successfully completed.

No configurations to display.

Server access logging

Log requests for access to your bucket. Learn more

Server access logging
Disabled

AWS CloudTrail data events

Configure CloudTrail data events to log Amazon S3 object-level API operations in the CloudTrail console. Learn more

Name	Access

No data events

No data events to display.

Event notifications (1)

Send a notification when specific events occur in your bucket. Learn more

□	Name	Event types	Filters	Destination type	Destination
□	S3PutEvent	Put	-	SNS topic	S3toEC2Topic

Amazon EventBridge

For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. Learn more or see EventBridge pricing

Send notifications to Amazon EventBridge for all events in this bucket
Off

Transfer acceleration

Use an accelerated endpoint for faster data transfers. Learn more

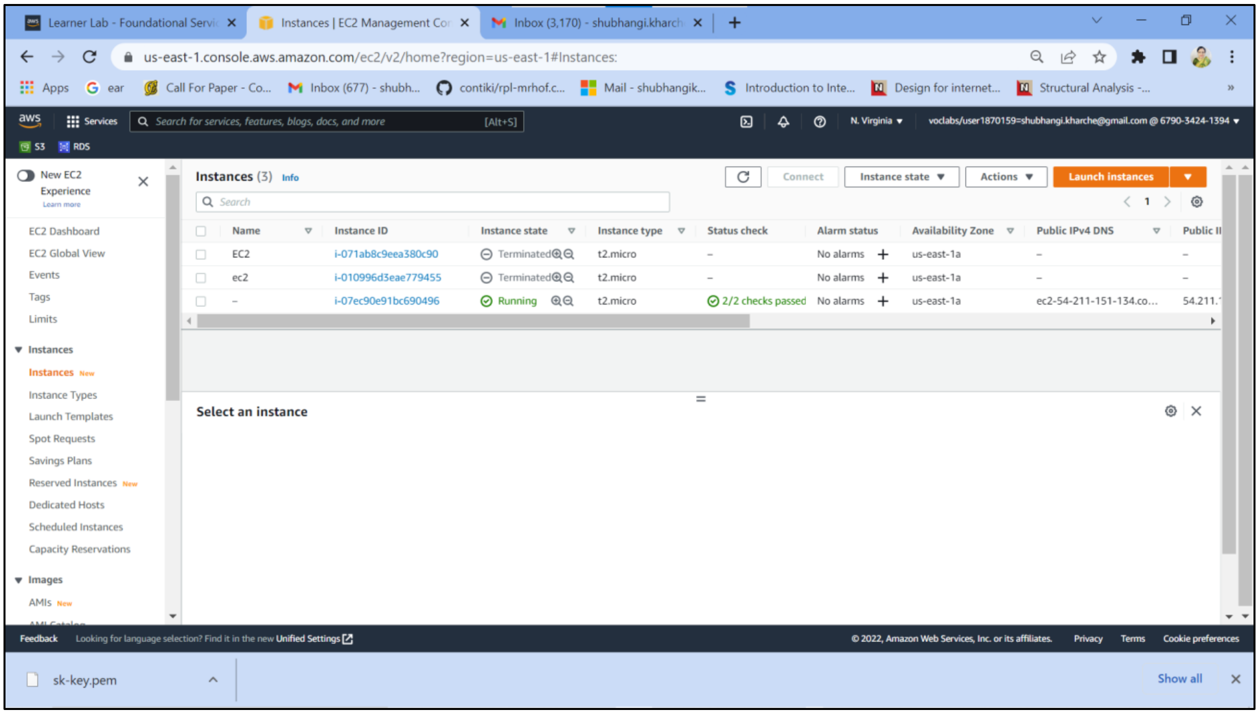
Transfer acceleration

Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Fig 4 (c): **Event Configuration details Screen-3**

Step 2: Run the custom program in the EC2 instance

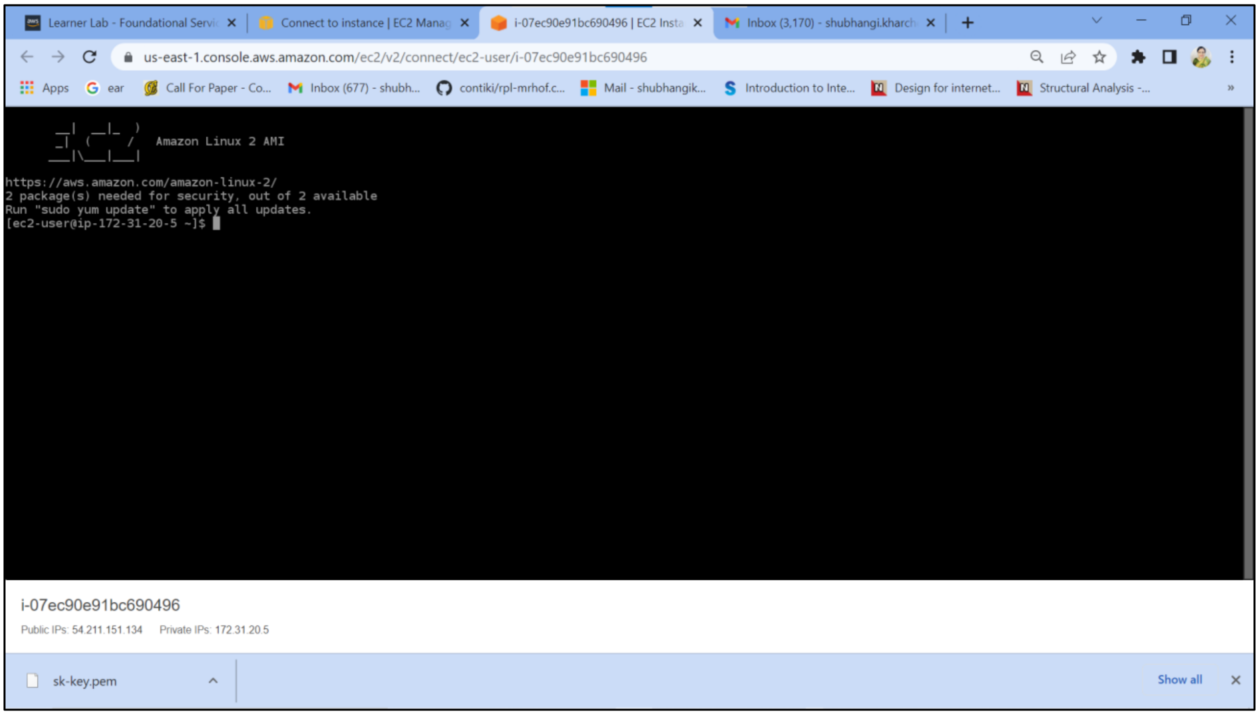
Step number	a
Step name	Creation of the EC2 instance and RDS instance
Instructions	1) Navigate to EC2 -> Instances 2) Create an EC2 instance with the following parameters AMI : Amazon Linux 2 VPC : Default Security group : Ports 22 and 8080 should be opened 3) Navigate to RDS 4) Create an RDS instance with the following parameters: Engine type : MySql Template : Dev/Test Set the username and password as required DB Instance class : Burstable Instance type : t3.micro Public Access : Yes VPC Security group : Create New () Under Additional Configuration, add an initial database name. Take note of this name as it will be required later. Uncheck “Enable Enhanced Monitoring” Ensure that the security group created by the RDS deployment has port 3306 open for all incoming connections from all sources.
Expected screenshots	1) List of instances after creation of EC2 instance 2) List of RDS instances


Learner Lab - Foundational Servi
Instances | EC2 Management Co
Inbox (3,170) - shubhangi.kharch
us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1870159=shubhangi.kharche@gmail.com @ 6790-3424-1394
S3
RDS
New EC2 Experience
Learn more
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
Connect
Instance state
Actions
Launch instances
Search
Select an instance
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
sk-key.pem
Show all

Instances (3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public II
EC2	i-071ab8c9eea380c90	Terminated	t2.micro	–	No alarms	us-east-1a	–	–
ec2	i-010996d3eae779455	Terminated	t2.micro	–	No alarms	us-east-1a	–	–
–	i-07ec90e91bc690496	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-211-151-134.co...	54.211.

Fig 5: List of EC2 instances after creation

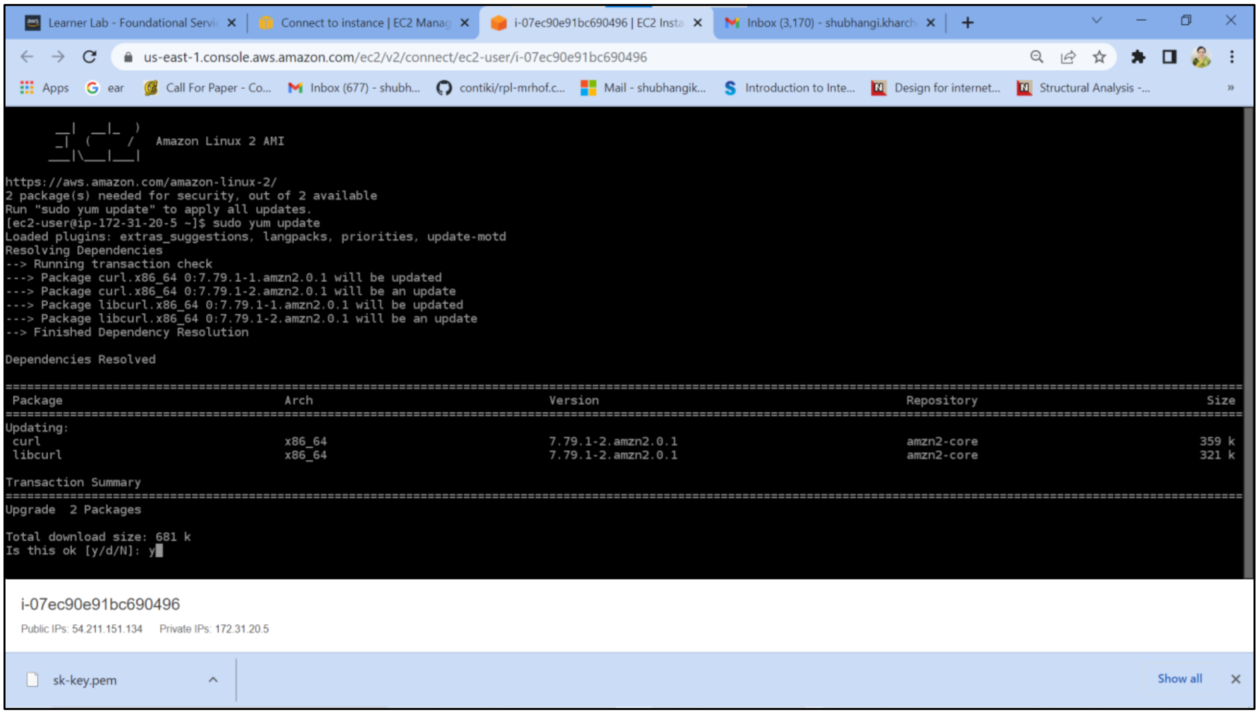

Learner Lab - Foundational Servi
Connect to instance | EC2 Manag
i-07ec90e91bc690496 | EC2 Insta
Inbox (3,170) - shubhangi.kharch
us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-07ec90e91bc690496
sk-key.pem
Show all

```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-20-5 ~]$ 
```

i-07ec90e91bc690496

Public IPs: 54.211.151.134 Private IPs: 172.31.20.5

Fig 5 (a): successful connection to EC2 instance (optional screenshot)

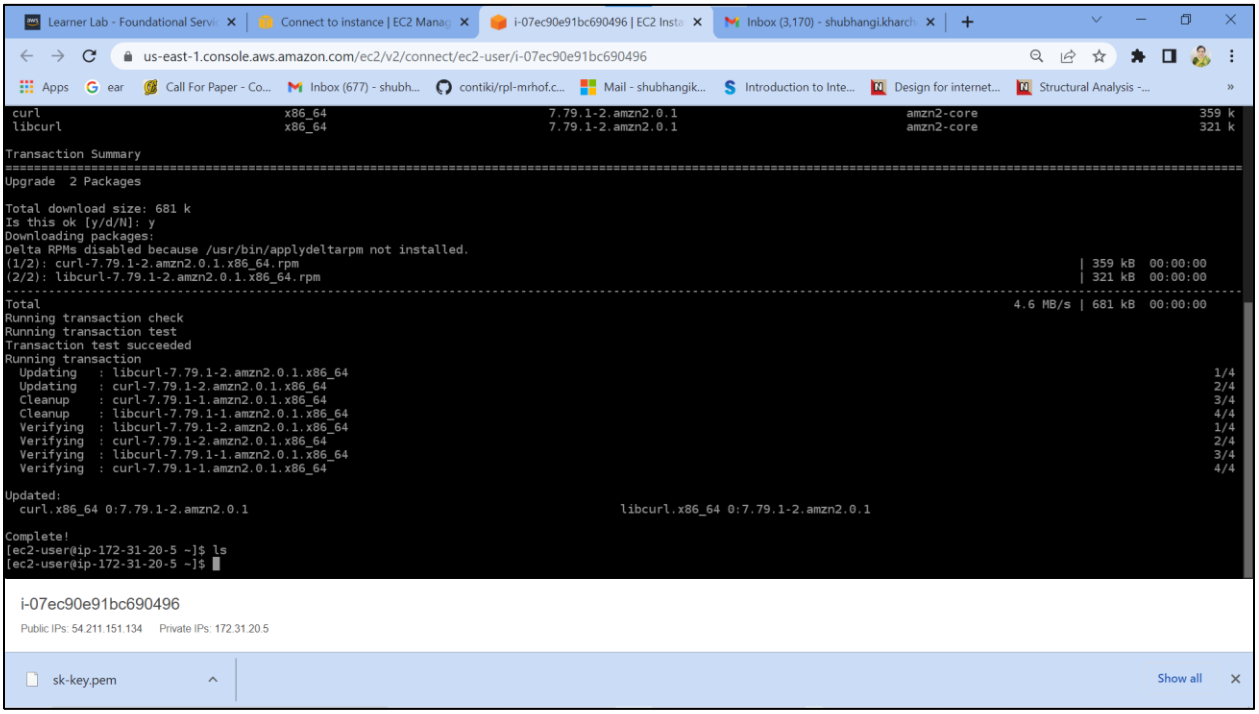

Learner Lab - Foundational Servic
Connect to instance | EC2 Manag
i-07ec90e91bc690496 | EC2 Insta
Inbox (3,170) - shubhangi.kharch
us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-07ec90e91bc690496
Apps
ear
Call For Paper - Co...
Inbox (677) - shubh...
contiki/rpl-mrhof.c...
Mail - shubhangik...
Introduction to Inte...
Design for internet...
Structural Analysis -...
sk-key.pem
Show all

```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-20-5 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package curl.x86_64 0:7.79.1-1.amzn2.0.1 will be updated
---> Package curl.x86_64 0:7.79.1-2.amzn2.0.1 will be an update
---> Package libcurl.x86_64 0:7.79.1-1.amzn2.0.1 will be updated
---> Package libcurl.x86_64 0:7.79.1-2.amzn2.0.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

============================================================================================================================================================================
 Package                               Arch                                 Version                                           Repository                                Size
============================================================================================================================================================================
Updating:
 curl                                  x86_64                               7.79.1-2.amzn2.0.1                                amzn2-core                               359 k
 libcurl                               x86_64                               7.79.1-2.amzn2.0.1                                amzn2-core                               321 k

Transaction Summary
============================================================================================================================================================================
Upgrade  2 Packages

Total download size: 681 k
Is this ok [y/d/N]: y
```

i-07ec90e91bc690496

Public IPs: 54.211.151.134   Private IPs: 172.31.20.5

Fig 5 (b): successful connection to EC2 instance (optional screenshot)

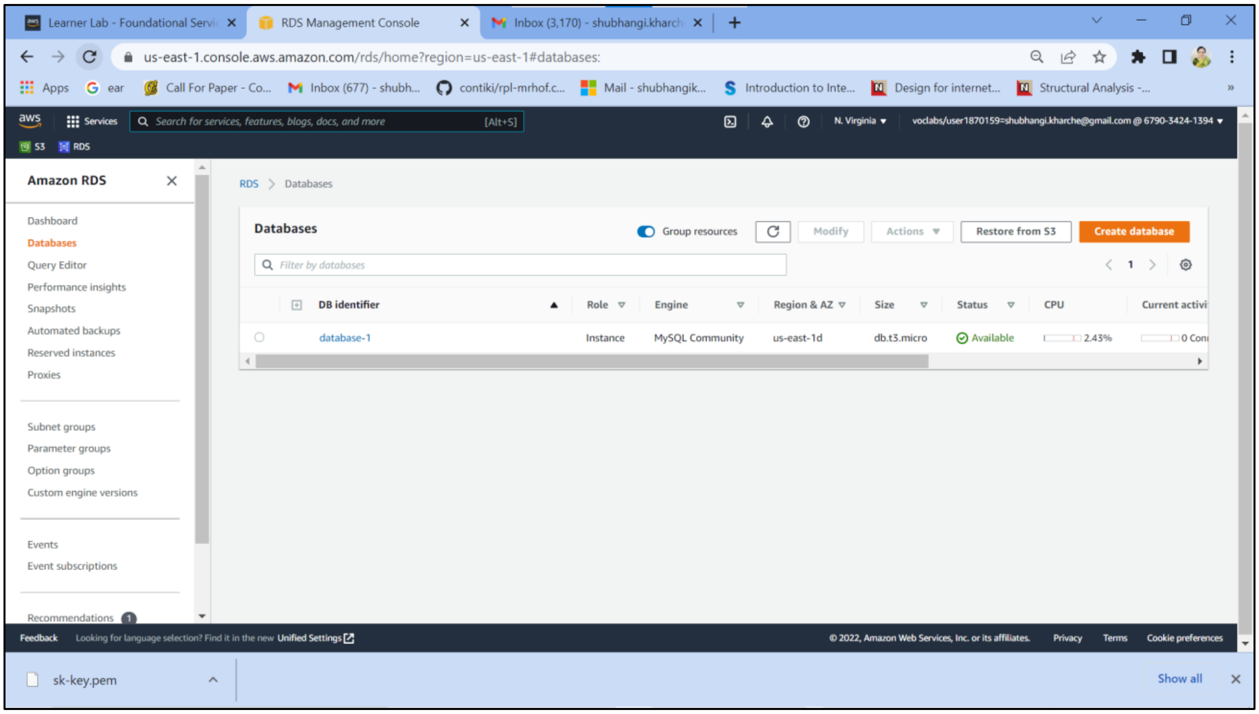

Learner Lab - Foundational Servic
Connect to instance | EC2 Manag
i-07ec90e91bc690496 | EC2 Insta
Inbox (3,170) - shubhangi.kharch
us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-07ec90e91bc690496
Apps
ear
Call For Paper - Co...
Inbox (677) - shubh...
contiki/rpl-mrhof.c...
Mail - shubhangik...
Introduction to Inte...
Design for internet...
Structural Analysis -...
sk-key.pem
Show all

```
 curl                                  x86_64                               7.79.1-2.amzn2.0.1                                amzn2-core                               359 k
 libcurl                               x86_64                               7.79.1-2.amzn2.0.1                                amzn2-core                               321 k

Transaction Summary
============================================================================================================================================================================
Upgrade  2 Packages

Total download size: 681 k
Is this ok [y/d/N]: y
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/2): curl-7.79.1-2.amzn2.0.1.x86_64.rpm                                                                                                             | 359 kB  00:00:00
(2/2): libcurl-7.79.1-2.amzn2.0.1.x86_64.rpm                                                                                                          | 321 kB  00:00:00
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                                                                        4.6 MB/s | 681 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : libcurl-7.79.1-2.amzn2.0.1.x86_64                                                                                                                       1/4
  Updating   : curl-7.79.1-2.amzn2.0.1.x86_64                                                                                                                          2/4
  Cleanup    : curl-7.79.1-1.amzn2.0.1.x86_64                                                                                                                          3/4
  Cleanup    : libcurl-7.79.1-1.amzn2.0.1.x86_64                                                                                                                       4/4
  Verifying  : libcurl-7.79.1-2.amzn2.0.1.x86_64                                                                                                                       1/4
  Verifying  : curl-7.79.1-2.amzn2.0.1.x86_64                                                                                                                          2/4
  Verifying  : libcurl-7.79.1-1.amzn2.0.1.x86_64                                                                                                                       3/4
  Verifying  : curl-7.79.1-1.amzn2.0.1.x86_64                                                                                                                          4/4

Updated:
  curl.x86_64 0:7.79.1-2.amzn2.0.1                                                      libcurl.x86_64 0:7.79.1-2.amzn2.0.1

Complete!
[ec2-user@ip-172-31-20-5 ~]$ ls
[ec2-user@ip-172-31-20-5 ~]$ 
```

i-07ec90e91bc690496

Public IPs: 54.211.151.134   Private IPs: 172.31.20.5

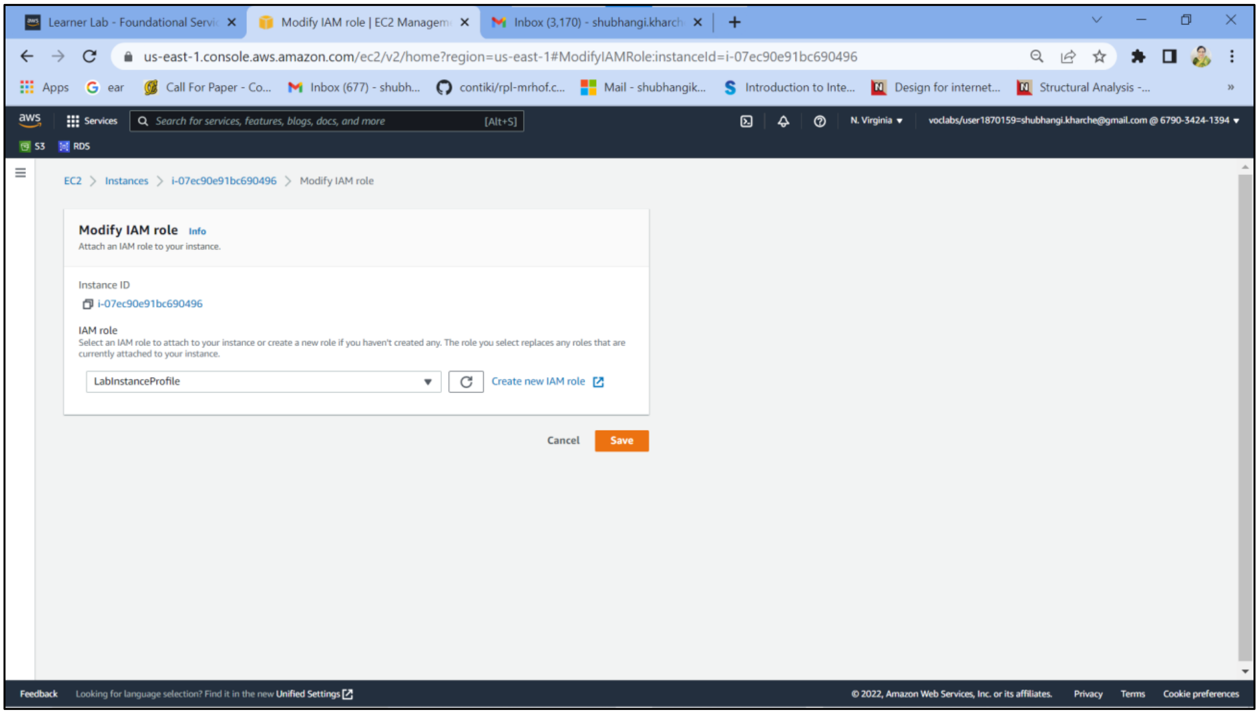
Fig 5 (c): successful connection to EC2 instance (optional screenshot)



DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activi
database-1	Instance	MySQL Community	us-east-1d	db.t3.micro	Available	2.43%	0 Con

Fig 6: List of RDS instances after creation

Step number	b
Step name	Assignment of IAM role for EC2 instance
Instructions	1) Navigate back to EC2- > Instances 2) Select the EC2 instance created in the previous step and select Actions-> Security -> Modify IAM role 3) Select the role LabInstanceProfile from the dropdown and click on Save
Expected screenshots	1) Modify IAM role screen


Learner Lab - Foundational Servic
Modify IAM role | EC2 Managem
Inbox (3,170) - shubhangi.kharch
Apps
ear
Call For Paper - Co...
Inbox (677) - shubh...
contiki/rpl-mrhof.c...
Mail - shubhangik...
Introduction to Inte...
Design for internet...
Structural Analysis -...
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
S3
RDS
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#ModifyIAMRole:instanceId=i-07ec90e91bc690496

voclabs/user1870159=shubhangi.kharche@gmail.com @ 6790-3424-1394

EC2 > Instances > i-07ec90e91bc690496 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

i-07ec90e91bc690496

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

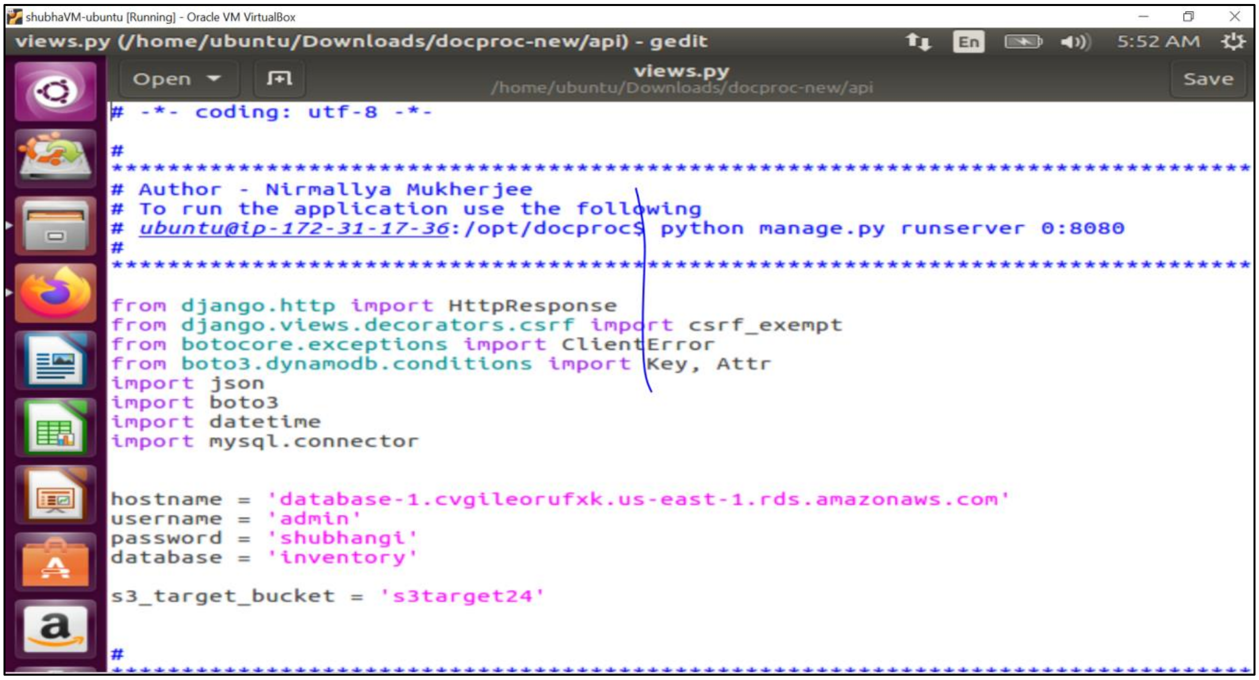
LabInstanceProfile

Create new IAM role

Cancel Save

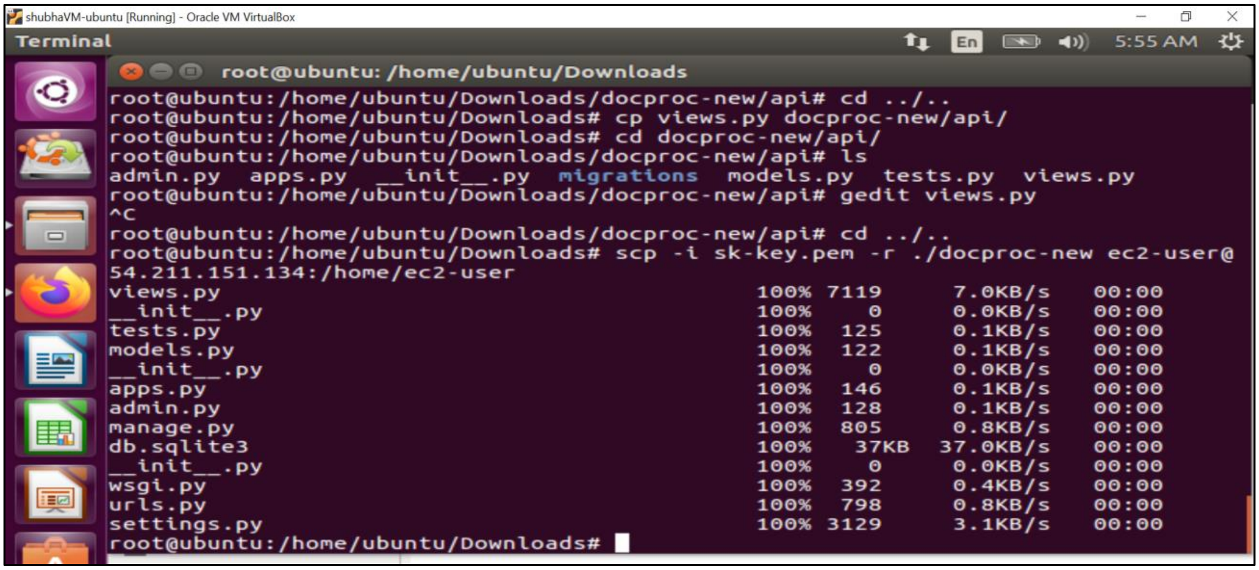
Fig 7: Modify IAM role screen

Step number	c	
Step name	Configuration and Uploading of custom program	
Instructions	1) Download the file **docproc-new.zip** on your machine 2) Unzip the downloaded file 3) Enter the unzipped folder and open the file views.py in the API folder using a text editor 4) In line number 19-24, modify the target bucket name to the one created in Step 2 (a) and modify the hostname, username, password and database variables to the values set while creating the RDS database and save the file 5) Copy the folder docproc-new to the home folder of the EC2 instance created in Step 3(a) using scp. Use the command given below *scp -i <pem> -r ./docproc-new ec2-user@<ip>:/home/ec2-user*	
Expected screenshots	1) Modifying of the views.py file to point to the target bucket	2)Copying the folder to the EC2 instance


shubhaVM-ubuntu [Running] - Oracle VM VirtualBox
views.py (/home/ubuntu/Downloads/docproc-new/api) - gedit
En
5:52 AM
Open
views.py
/home/ubuntu/Downloads/docproc-new/api
Save

```
# -*- coding: utf-8 -*-

#
*************************************************************************************************
# Author - Nirmallya Mukherjee
# To run the application use the following
# ubuntu@ip-172-31-17-36:/opt/docproc$ python manage.py runserver 0:8080
#
*************************************************************************************************

from django.http import HttpResponse
from django.views.decorators.csrf import csrf_exempt
from botocore.exceptions import ClientError
from boto3.dynamodb.conditions import Key, Attr
import json
import boto3
import datetime
import mysql.connector


hostname = 'database-1.cvgileorufxk.us-east-1.rds.amazonaws.com'
username = 'admin'
password = 'shubhangi'
database = 'inventory'

s3_target_bucket = 's3target24'


#
*************************************************************************************************
```

Fig 8: **Modifying of the views.py file to point to the target bucket**

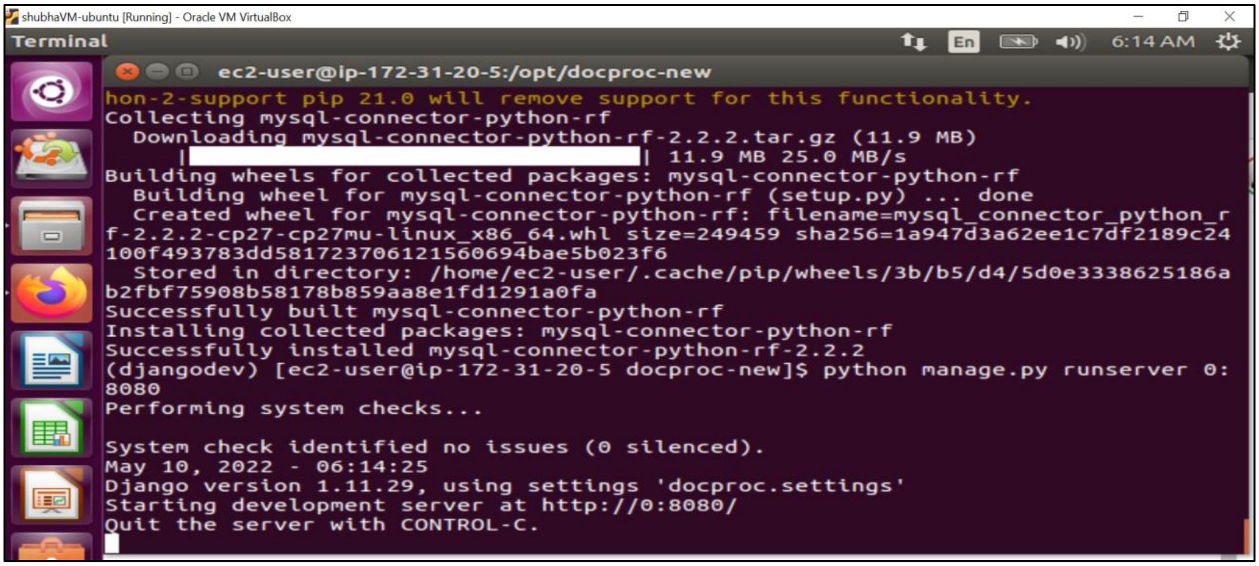

shubhaVM-ubuntu [Running] - Oracle VM VirtualBox
Terminal
En
5:55 AM
root@ubuntu: /home/ubuntu/Downloads

```
root@ubuntu:/home/ubuntu/Downloads/docproc-new/api# cd ../..
root@ubuntu:/home/ubuntu/Downloads# cp views.py docproc-new/api/
root@ubuntu:/home/ubuntu/Downloads# cd docproc-new/api/
root@ubuntu:/home/ubuntu/Downloads/docproc-new/api# ls
admin.py  apps.py  __init__.py  migrations  models.py  tests.py  views.py
root@ubuntu:/home/ubuntu/Downloads/docproc-new/api# gedit views.py
^C
root@ubuntu:/home/ubuntu/Downloads/docproc-new/api# cd ../..
root@ubuntu:/home/ubuntu/Downloads# scp -i sk-key.pem -r ./docproc-new ec2-user@
54.211.151.134:/home/ec2-user
views.py                                      100% 7119     7.0KB/s   00:00
__init__.py                                   100%    0     0.0KB/s   00:00
tests.py                                      100%  125     0.1KB/s   00:00
models.py                                     100%  122     0.1KB/s   00:00
__init__.py                                   100%    0     0.0KB/s   00:00
apps.py                                       100%  146     0.1KB/s   00:00
admin.py                                      100%  128     0.1KB/s   00:00
manage.py                                     100%  805     0.8KB/s   00:00
db.sqlite3                                    100%   37KB  37.0KB/s   00:00
__init__.py                                   100%    0     0.0KB/s   00:00
wsgi.py                                       100%  392     0.4KB/s   00:00
urls.py                                       100%  798     0.8KB/s   00:00
settings.py                                   100% 3129     3.1KB/s   00:00
root@ubuntu:/home/ubuntu/Downloads# 
```

Fig 9: **Copying the folder to the EC2 instance**

Step 3: Creation and Verification of SNS subscription and Generation of CSV file

Step number	a
Step name	Starting the EC2 custom program
Instructions	1) Log into the EC2 instance using SSH 2) Run the followng commands after successful SSH to start the server sudo cp -r docproc-new /opt sudo chown ec2-user:ec2-user -R /opt cd /opt/docproc-new sudo yum update sudo yum install python-pip -y python -m pip install --upgrade pip setuptools sudo pip install virtualenv virtualenv ~/.virtualenvs/djangodev source ~/.virtualenvs/djangodev/bin/activate pip install django pip install boto3 pip install mysql-connector-python-rf python manage.py runserver 0:8080 **Keep this terminal window open throughout the rest of the exercise**
Expected screenshots	1) Server in waiting state

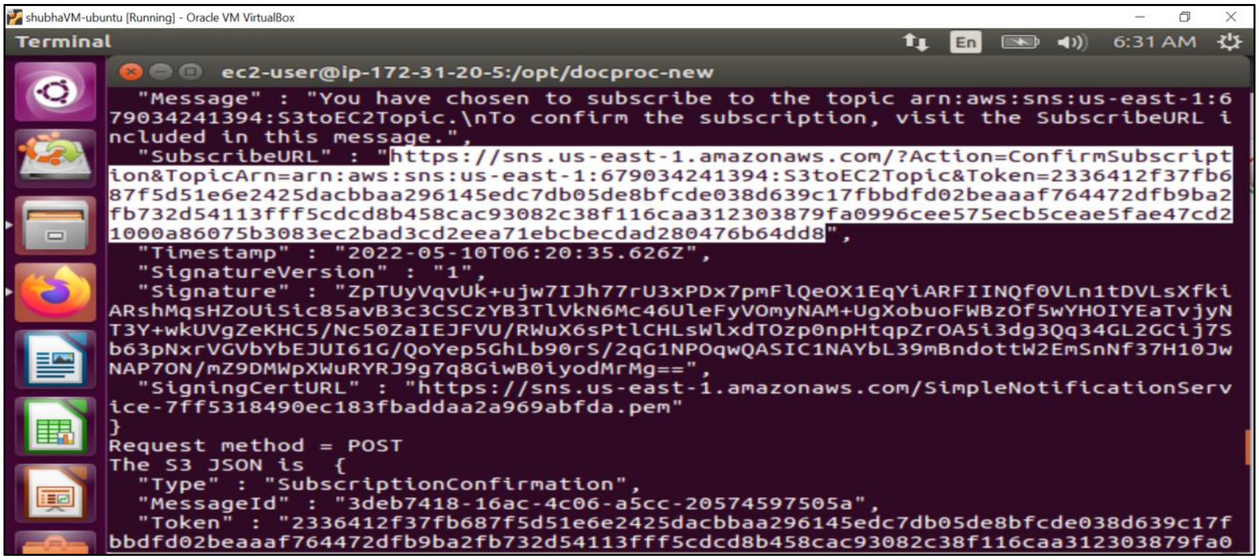


shubhaVM-ubuntu [Running] - Oracle VM VirtualBox
Terminal
En
6:14 AM
ec2-user@ip-172-31-20-5:/opt/docproc-new

```
hon-2-support pip 21.0 will remove support for this functionality.
Collecting mysql-connector-python-rf
  Downloading mysql-connector-python-rf-2.2.2.tar.gz (11.9 MB)
     |████████████████████████████████| 11.9 MB 25.0 MB/s
Building wheels for collected packages: mysql-connector-python-rf
  Building wheel for mysql-connector-python-rf (setup.py) ... done
  Created wheel for mysql-connector-python-rf: filename=mysql_connector_python_r
f-2.2.2-cp27-cp27mu-linux_x86_64.whl size=249459 sha256=1a947d3a62ee1c7df2189c24
100f493783dd581723706121560694bae5b023f6
  Stored in directory: /home/ec2-user/.cache/pip/wheels/3b/b5/d4/5d0e3338625186a
b2fbf75908b58178b859aa8e1fd1291a0fa
Successfully built mysql-connector-python-rf
Installing collected packages: mysql-connector-python-rf
Successfully installed mysql-connector-python-rf-2.2.2
(djangodev) [ec2-user@ip-172-31-20-5 docproc-new]$ python manage.py runserver 0:
8080
Performing system checks...

System check identified no issues (0 silenced).
May 10, 2022 - 06:14:25
Django version 1.11.29, using settings 'docproc.settings'
Starting development server at http://0:8080/
Quit the server with CONTROL-C.
```

Fig 10: Server in waiting state

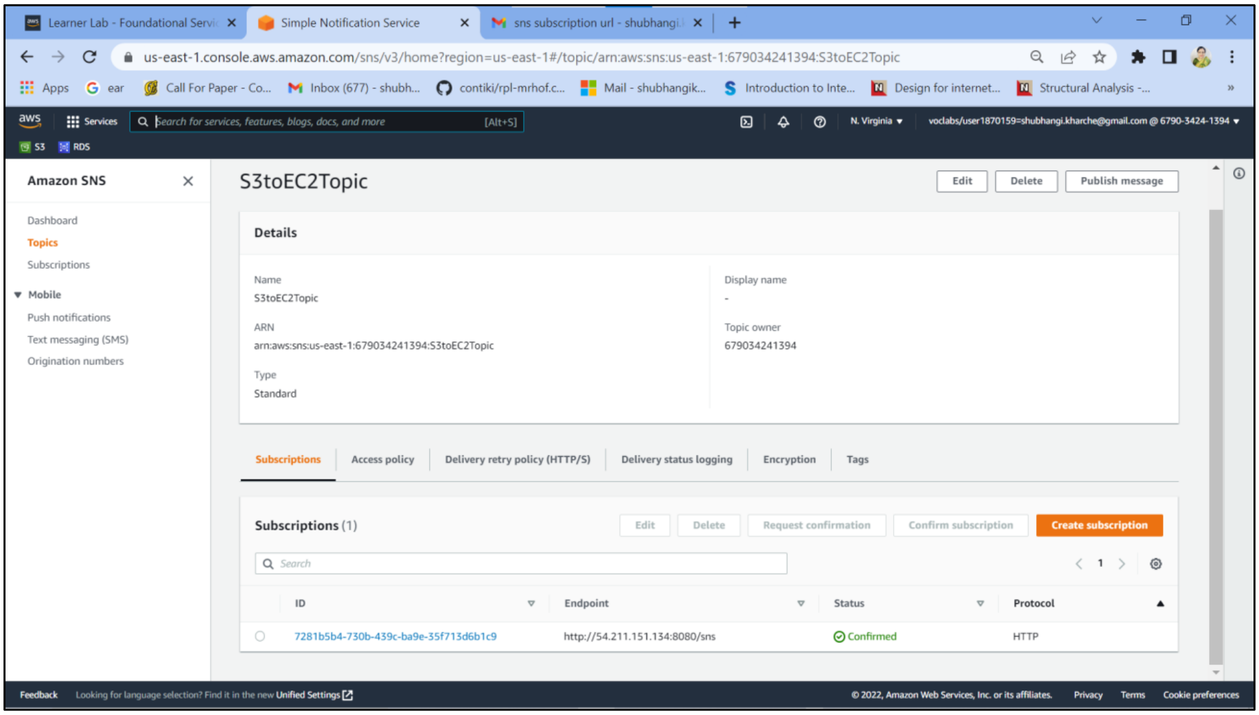
Step number	b
Step name	Creation of SNS subscription
Instructions	1) Navigate to SNS in the AWS Console and select the topic S3ToEC2Topic 2) Click on Create Subscription 3) Enter the following details Protocol : HTTP Endpoint : http://<host>:8080/sns where <host> in the public IP of the EC2 instance Click on Create Subscription 4) In the EC2 terminal window, look for the field "SubscribeURL" and copy the entire link given **Note: If a message is seen "ValueError: No JSON object could be decoded", it can be safely ignored** 5) Paste that link into a browser window to verify the SNS subscription (Ignore any messages received in the web browser)
Expected screenshots	1) Subscription URL in EC2 terminal Window



shubhaVM-ubuntu [Running] - Oracle VM VirtualBox
Terminal
En
6:31 AM
ec2-user@ip-172-31-20-5:/opt/docproc-new

```
  "Message" : "You have chosen to subscribe to the topic arn:aws:sns:us-east-1:6
79034241394:S3toEC2Topic.\nTo confirm the subscription, visit the SubscribeURL i
ncluded in this message.",
  "SubscribeURL" : "https://sns.us-east-1.amazonaws.com/?Action=ConfirmSubscript
ion&TopicArn=arn:aws:sns:us-east-1:679034241394:S3toEC2Topic&Token=2336412f37fb6
87f5d51e6e2425dacbbaa296145edc7db05de8bfcde038d639c17fbbdfd02beaaaf764472dfb9ba2
fb732d54113fff5cdcd8b458cac93082c38f116caa312303879fa0996cee575ecb5ceae5fae47cd2
1000a86075b3083ec2bad3cd2eea71ebcbecdad280476b64dd8",
  "Timestamp" : "2022-05-10T06:20:35.626Z",
  "SignatureVersion" : "1",
  "Signature" : "ZpTUyVqvUk+ujw7IJh77rU3xPDx7pmFlQeOX1EqYiARFIINQf0VLn1tDVLsXfki
ARshMqsHZoUiSic85avB3c3CSCzYB3TlVkN6Mc46UleFyVOmyNAM+UgXobuoFWBzOf5wYHOIYEaTvjyN
T3Y+wkUVgZeKHC5/Nc50ZaIEJFVU/RWuX6sPtlCHLsWlxdTOzp0npHtqpZrOA5i3dg3Qq34GL2GCij7S
b63pNxrVGVbYbEJUI61G/QoYep5GhLb90rS/2qG1NPOqwQASIC1NAYbL39mBndottW2EmSnNf37H10Jw
NAP7ON/mZ9DMWpXWuRYRJ9g7q8GiwB0iyodMrMg==",
  "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationServ
ice-7ff5318490ec183fbaddaa2a969abfda.pem"
}
Request method = POST
The S3 JSON is  {
  "Type" : "SubscriptionConfirmation",
  "MessageId" : "3deb7418-16ac-4c06-a5cc-20574597505a",
  "Token" : "2336412f37fb687f5d51e6e2425dacbbaa296145edc7db05de8bfcde038d639c17f
bbdfd02beaaaf764472dfb9ba2fb732d54113fff5cdcd8b458cac93082c38f116caa312303879fa0
```

Fig 11: **Subscription URL in EC2 terminal Window**



S3toEC2Topic

Details

Name
S3toEC2Topic

ARN
arn:aws:sns:us-east-1:679034241394:S3toEC2Topic

Type
Standard

Display name
-

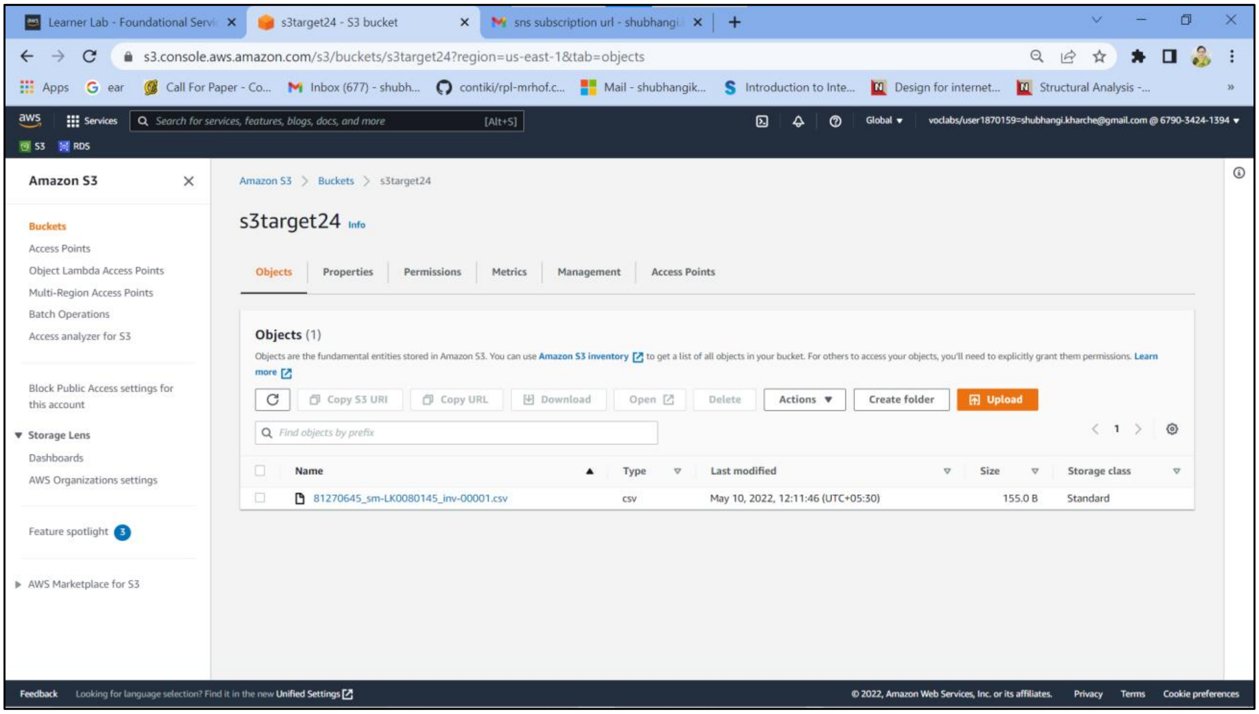
Topic owner
679034241394

Subscriptions (1)

	ID	Endpoint	Status	Protocol
○	7281b5b4-730b-439c-ba9e-35f713d6b1c9	http://54.211.151.134:8080/sns	✓ Confirmed	HTTP

Fig 11 (a): **Subscription confirmed in sns (optional screenshot)**

Step number	c
Step name	Generation of CSV file
Instructions	1) Download the file **docproc-invoice.txt** provided with this workbook 2) Navigate to S3 in the AWS Console 3) Upload the sample invoice file to the source S3 bucket using the default options 4) Verify that a CSV file is generated in the target S3 bucket. This may take a few minutes 5) (Optional) Login to the RDS instance using your preferred MySQL client and check the table created inside the specified database.
Expected screenshots	1) Generated CSV file in the target S3 bucket



s3.console.aws.amazon.com/s3/buckets/s3target24?region=us-east-1&tab=objects

Amazon S3 > Buckets > s3target24

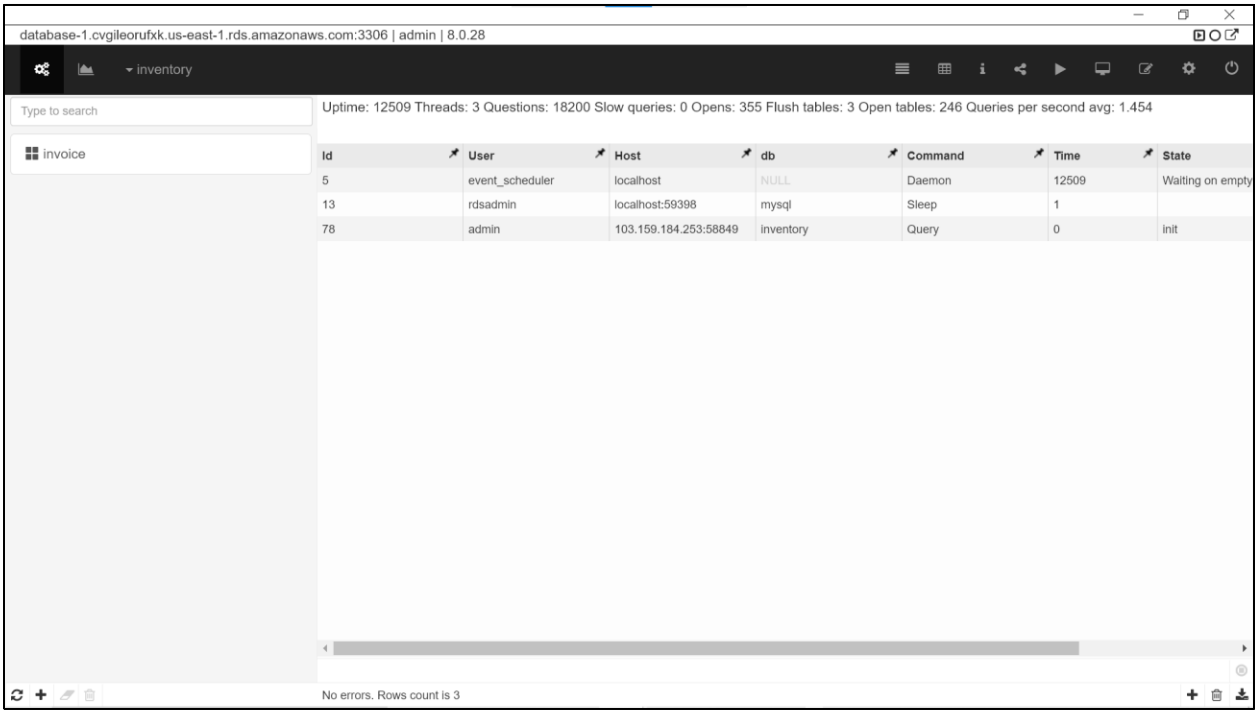
s3target24 Info

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
81270645_sm-LK0080145_inv-00001.csv	csv	May 10, 2022, 12:11:46 (UTC+05:30)	155.0 B	Standard

Fig 12: Generated CSV file in the target S3 bucket



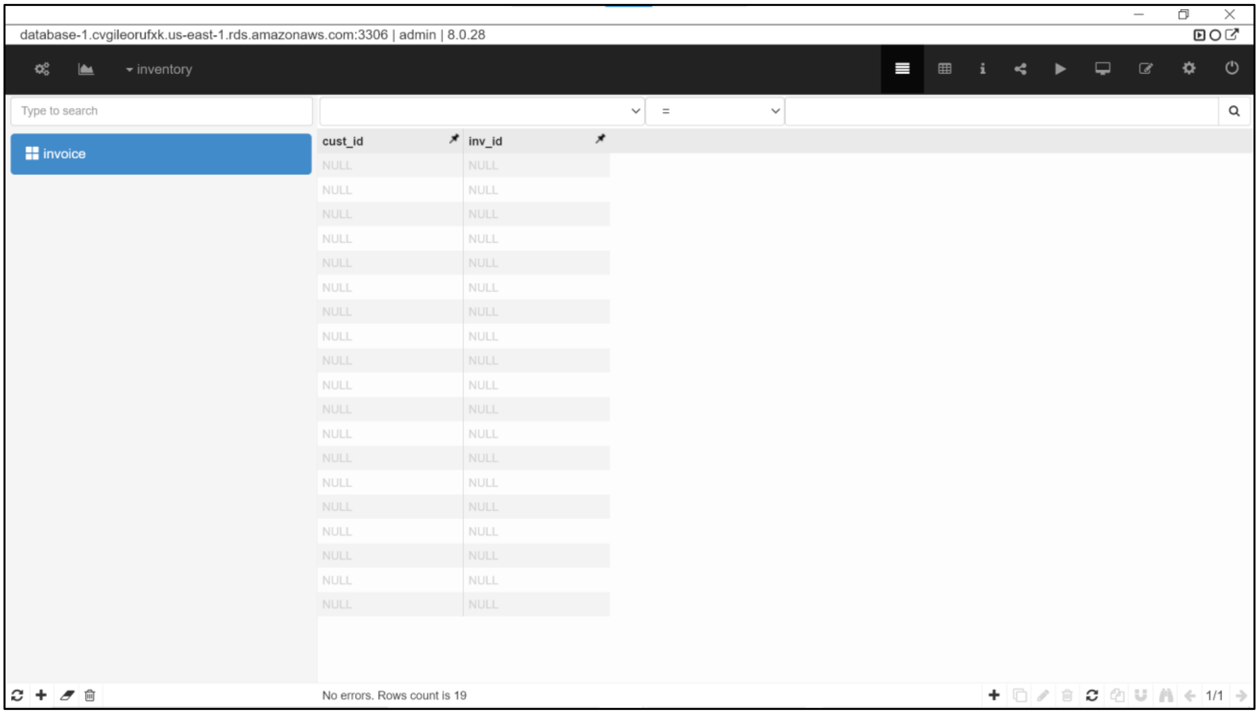
database-1.cvgileorufxk.us-east-1.rds.amazonaws.com:3306 | admin | 8.0.28

Uptime: 12509 Threads: 3 Questions: 18200 Slow queries: 0 Opens: 355 Flush tables: 3 Open tables: 246 Queries per second avg: 1.454

Id	User	Host	db	Command	Time	State
5	event_scheduler	localhost	NULL	Daemon	12509	Waiting on empty
13	rdsadmin	localhost:59398	mysql	Sleep	1	
78	admin	103.159.184.253:58849	inventory	Query	0	init

No errors. Rows count is 3

Fig 12 (a): Table created in RDS database (optional screenshot)



database-1.cvgileorufxk.us-east-1.rds.amazonaws.com:3306 | admin | 8.0.28

cust_id	inv_id
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL

No errors. Rows count is 19

Fig 12 (b): Invoice Table created in RDS database (optional screenshot)

Answer the following questions

Q1 Which of the following properties of an AWS resource is sufficient and necessary to uniquely identify it across all of AWS?

a) ARN

b) Region and ARN

c) ARN and Account number

d) Depends on the resource used

Enter your answer here ARN

Q2 Which of the following step numbers in Step 1 allowed S3 to publish to the SNS topic created?

a) 1(a)

b) 1(c)

c) 1(d)

d) 1(b)

Enter your answer here b) 1 (c)

Q3 Which port is being used by SNS to send the notification to the custom program?

a) 8081

b) 80

c) 8080

d) 8065

Enter your answer here c) 8080

Q4 How many IAM roles can be attached to an EC2 instance at a time?

a) 2

b) 3

c) 1

d) Depends on the policies required

Enter your answer here c) 1

Q5 As a product manager, how would you describe the benefits of this architecture to an client, as compared to an equivalent on-premises architecture?

Following are the benefits of the architecture to a client as compared to an equivalent on-premises architecture:

1) Cloud architecture provides fully managed services.
2) No overhead of maintaining storage, compute, database, MySQL, API, python scripts etc. on premises.
3) .txt can be uploaded from anywhere to source bucket and .csv can be accessed from anywhere from target bucket.
4) Message is placed in SNS topic in response to event triggered in source s3 bucket.
5) EC2 gets the message from s3 source bucket with the help of custom program that subscribes to the SNS topic.
6) The s3 source bucket content is read, converted to CSV with the help of API and RDS database.
7) CSV record is written in s3 target bucket as an object.
8) Steps 3 to 7 are fully managed; the client need not maintain any of the required services on premises.
9) The client will save on hosting and managing the infrastructure on premises.