

PGPCC
Project Implementation
Building an Automated Business Process
using Managed Services on a Public Cloud
Phase 2– Implementation
--Mahesh Jasti

Contents

Phase-1 Architecture

Section 1: Objective of the Project	Pg.3
Section 2: Scope of the Project.....	Pg.3
Section 3: Implementation Architecture.....	Pg.4

Phase-2 Implementation

Section 4: High level steps to implement the project...	5
Section 5: Screenshots from AWS console.....	5-28
Section 6: Lessons learnt / Observations.....	29

Section 1: Objective of the Project

The objective of this project is to create an automated, event based real time process that does not have these limitations. Data should flow rapidly from the source to the destination.

Section 2: Scope of the Project

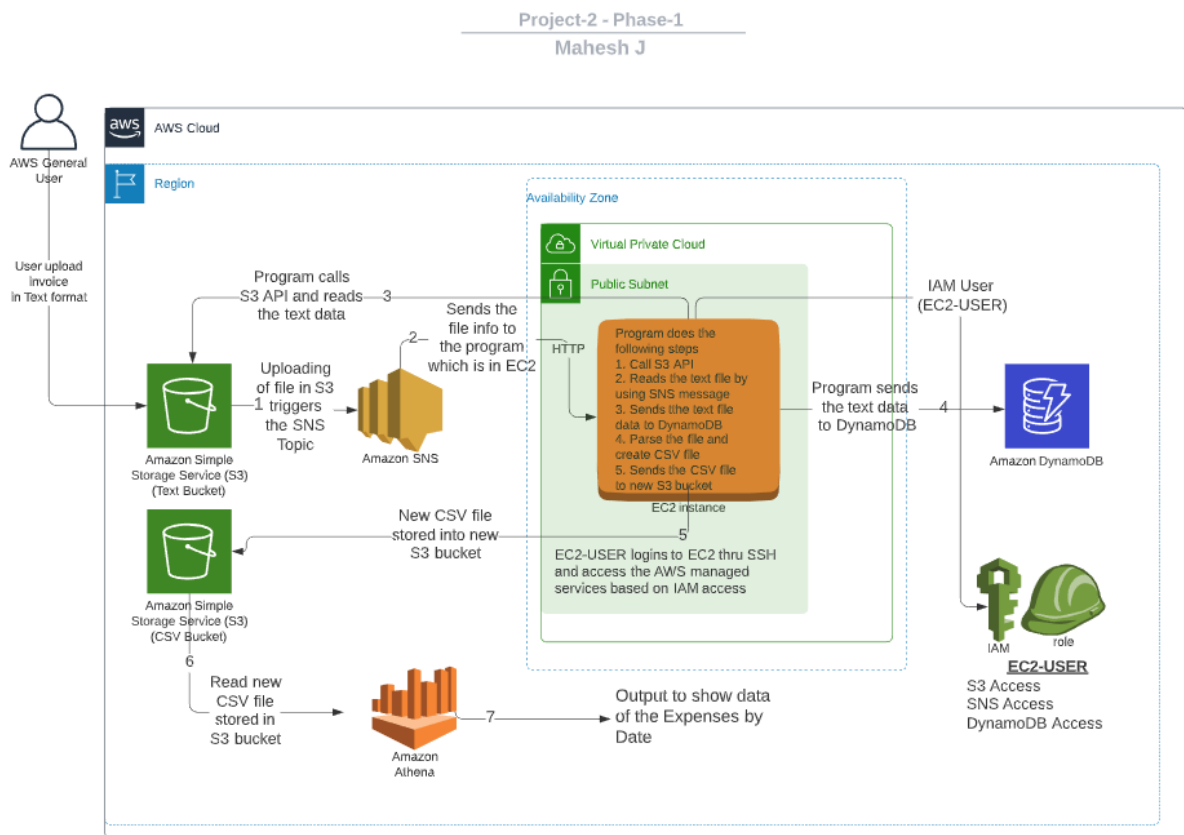
Create a solution architecture diagram based on the below points..

- 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 along with saving the original record in DynamoDB.
- 5.The program will use S3 API to write CSV record to destination S3 bucket as new S3 object.

6. Athena is used to query the CSV file (query to show aggregated expenses grouped by date).

Section 3: Implementation Architecture

The following picture outlines the implementation architecture for the project.



I have put the numbers from 1 to 7 to implement the flow. Below is the explanation for each step.

1. When user uploads the text file in S3 then it triggers the SNS topic.
2. SNS sends the notification to EC2 program.

3. EC2 program checks for the IAM access and then program calls the S3 api.
4. EC2 Program reads the data from text file and loads into dynamo DB.
5. EC2 program parses the data and creates the CSV file and store into new S3 bucket.
6. Athena reads the CSV file from the new S3 bucket.
7. Run the SQL query to show the result.

Section 4: High level steps to implement the project

1. Create the IAM role for EC2 Instance (S3 and DynamoDB)
2. Create the VPC and Subnet and initiate an EC2 instance
3. Login to the instance and setup the softwares and program as per the given instructions
4. Setup S3 and SNS event trigger
5. Verify DynamoDB after input file is loaded into S3
6. Setup the Athena by using S3 target CSV file

Section 5: Screenshots from AWS console

IAM Role: Create a role in IAM to have EC2 access the S3 and Dynamo DB.

PGPCC – Project Implementation Task

aws


Services ▾ Resource Groups ▾


🔔 MaheshJasti ▾ Global ▾ Support ▾


Create role


1 2 3 4

Select type of trusted entity

 **AWS service**
EC2, Lambda and others

 **Another AWS account**
Belonging to you or 3rd party

 **Web identity**
Cognito or any OpenID provider

 **SAML 2.0 federation**
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway

CodeGuru

ElastiCache

Kinesis

RoboMaker

AWS Backup

CodeStar Notifications

Elastic Beanstalk

Lake Formation

S3

AWS Chatbot

Comprehend

Elastic Container Service

Lambda

SMS

AWS Support

Config

Elastic Transcoder

Lex

SNS

* Required

Cancel

Next: Permissions

Select S3 full access and DynamoDB full access..

aws

Services ▾ Resource Groups ▾

🔔 MaheshJasti ▾ Global ▾ Support ▾

Create role

1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.









Create policy

↺

Filter policies ▾

Q dyn

Showing 8 results

	Policy name ▾	Used as
<input checked="" type="checkbox"/>	▶  AmazonDynamoDBFullAccess	Permissions policy (1)
<input type="checkbox"/>	▶  AmazonDynamoDBFullAccesswithDataPipeline	None
<input type="checkbox"/>	▶  AmazonDynamoDBReadOnlyAccess	None
<input type="checkbox"/>	▶  AWSApplicationAutoscalingDynamoDBTablePolicy	None
<input type="checkbox"/>	▶  AWSLambdaDynamoDBExecutionRole	None
<input type="checkbox"/>	▶  AWSLambdaInvocation-DynamoDB	None
<input type="checkbox"/>	▶  DynamoDBCloudWatchContributorInsightsServiceRolePolicy	None
<input type="checkbox"/>	▶  DynamoDBReplicationServiceRolePolicy	None

▶ Set permissions boundary

* Required

Cancel

Previous

Next: Tags

PGPCC – Project Implementation Task

aws

Services

Resource Groups

★

🔔

MaheshJasti

Create role

1 2 3 4

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Name"/>	<input type="text" value="EC2-Project2"/>	✕
<input type="text" value="Add new key"/>	<input type="text"/>	

You can add 49 more tags.

Enter the tag name and Role name and click on Create Role button.

aws

Services

Resource Groups

★

🔔

MaheshJasti

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name*

Use alphanumeric and '+', '@', '-', '_' characters. Maximum 64 characters.

Role description

Maximum 1000 characters. Use alphanumeric and '+', '@', '-', '_' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies [AmazonS3FullAccess](#) [AmazonDynamoDBFullAccess](#)

Permissions boundary Permissions boundary is not set

The new role will receive the following tag

Key	Value

* Required

[Cancel](#) [Previous](#) [Create role](#)

Create a VPC

aws Services Resource Groups MaheshJasti

VPCs > Create VPC

Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an IPv6 CIDR block with the VPC.

Name tag

IPv4 CIDR block*

IPv6 CIDR block

- ☒ No IPv6 CIDR Block
- ☐ Amazon provided IPv6 CIDR block
- ☐ IPv6 CIDR owned by me

Tenancy

VPCs > Create VPC

Create VPC

✓ The following VPC was created:

VPC ID `vpc-001a119b0f3348e9c`

Close

Now create subnets.

aws Services Resource Groups MaheshJasti Oregon Support

Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag

VPC*

Availability Zone

VPC CIDRs	CIDR	Status	Status Reason
	10.40.0.0/16	associated	

IPv4 CIDR block*

* Required

Cancel Create

Create Internet Gateway and assign to the new VPC.

Internet gateway igw-076410745b2524737 successfully attached to vpc-001a119b0f3348e9c

VPC > Internet gateways > igw-076410745b2524737

igw-076410745b2524737 / project2-IGWY

Details Info

Internet gateway ID igw-076410745b2524737	State Attached	VPC ID vpc-001a119b0f3348e9c VPC-Project2	Owner 554932232731
----------------------------------------------	-------------------	------------------------------------------------	-----------------------

Tags

Search tags

Key	Value
Name	project2-IGWY

Edit the route table and add the Internet Gateway and also associate the Subnet..

Create route table Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
Project2-Route Table	rtb-0dd5104be84c4507b	subnet-0e749432edc5eaaa	-	Yes	vpc-001a119b0f3348e9c ...	55493223
Default Route Table	rtb-1288d469	-	-	Yes	vpc-71dead09	55493223

Route Table: rtb-0dd5104be84c4507b

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

Destination	Target	Status	Propagated
10.40.0.0/16	local	active	No
0.0.0.0/0	igw-076410745b2524737	active	No

PGPCC – Project Implementation Task

The screenshot shows the AWS Management Console interface. On the left is the navigation menu with options like 'VPC Dashboard', 'Subnets', 'Route Tables', 'Internet Gateways', etc. The main content area displays a list of route tables. Below this, the details for 'Route Table: rtb-0dd5104be84c4507b' are shown, specifically the 'Subnet Associations' tab. A table lists the association for 'subnet-0e7494323edc5eaaa' with an IPv4 CIDR of '10.40.1.0/24'.

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
Project2-Route Table	rtb-0dd5104be84c4507b	subnet-0e7494323edc5eaaa	-	Yes	vpc-001a119b0f3348e9c	55493223
Default Route Table	rtb-1288d469	-	-	Yes	vpc-71dead09	55493223

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0e7494323edc5eaaa	10.40.1.0/24	-

Enable auto-assign public IPV4 address

The screenshot shows the 'Modify auto-assign IP settings' page. It includes a description of the setting and a form where the 'Auto-assign IPv4' checkbox is checked. Below the form, there is a note about required fields.

Subnets > Modify auto-assign IP settings

Modify auto-assign IP settings

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched in this subnet. You can override the auto-assign IP settings for an instance at launch time.

Subnet ID: subnet-0e7494323edc5eaaa

Auto-assign IPv4 ☒ Enable auto-assign public IPv4 address ⓘ

* Required

Now create the EC2 instance... (Ubuntu Instance)..

PGPCC – Project Implementation Task

The screenshot shows the AWS console's 'Step 1: Choose an Amazon Machine Image (AMI)' screen. The navigation bar at the top includes 'Services', 'Resource Groups', and a user profile 'MaheshJasti' in 'Oregon'. The progress bar shows steps: 1. Choose AMI (active), 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review. The main content area lists several AMIs with their details and 'Select' buttons:

- Red Hat Enterprise Linux 8 (HVM), SSD Volume Type** - ami-02f147dfb8be58a10 (64-bit x86) / ami-04b741928ba3831b2 (64-bit Arm)
- SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type** - ami-0d218397f35e41108 (64-bit x86) / ami-060724914319ccd9c (64-bit Arm)
- Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-003634241a8fcdcd (64-bit x86) / ami-0f494dc39ae364f3d (64-bit Arm)

Each entry includes details like 'Free tier eligible', 'Root device type: ebs', 'Virtualization type: hvm', and 'ENA Enabled: Yes'. A 'Cancel and Exit' link is at the top right. At the bottom, there is a promotional banner for Amazon RDS.

Select t2.micro instance and click “Configure Instance details” button..

The screenshot shows the AWS console's 'Step 2: Choose an Instance Type' screen. The navigation bar and progress bar are consistent with the previous step. The main content area displays a table of instance types, filtered by 'All instance types' and 'Current generation'. The 't2.micro' instance is selected and highlighted in green, with a 'Free tier eligible' tag. The table columns include Family, Type, vCPUs, Memory (GiB), Instance Storage (GiB), EBS-Optimized Available, Network Performance, and IPv6 Support.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Instance Details'.

Select the VPC and Subnet and click on “Add Storage” button.

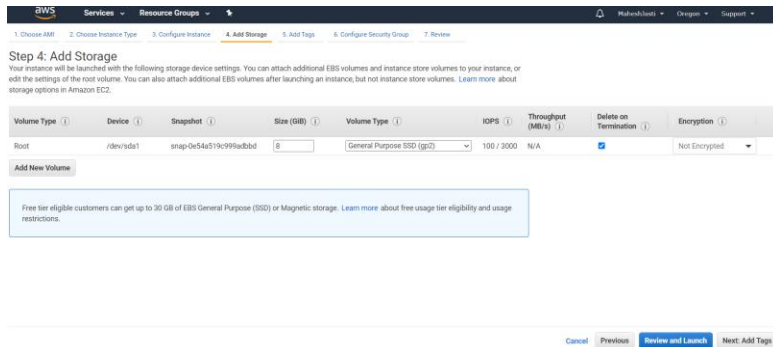
The screenshot shows the AWS console's 'Step 3: Configure Instance Details' screen. The navigation bar and progress bar are consistent. The main content area contains various configuration options for the instance:

- Number of Instances:** 1 (with a 'Launch into Auto Scaling Group' link)
- Purchasing option:** ☐ Request Spot Instances
- Network:** vpc-001a119b0f3348e9c | VPC-Project2 (with a 'Create new VPC' link)
- Subnet:** subnet-0e7494323edc5eaaa | Public Subnet | us-west (with a 'Create new subnet' link)
- Auto-assign Public IP:** Use subnet setting (Enable)
- Placement group:** ☐ Add instance to placement group
- Capacity Reservation:** Open
- IAM role:** None (with a 'Create new IAM role' link)
- Shutdown behavior:** Stop
- Stop - Hibernate behavior:** ☐ Enable hibernation as an additional stop behavior
- Enable termination protection:** ☐ Protect against accidental termination
- Monitoring:** ☐ Enable CloudWatch detailed monitoring

At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Storage'.

No changes to Add Storage screen.. Move to Tags screen..

PGPCC – Project Implementation Task



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

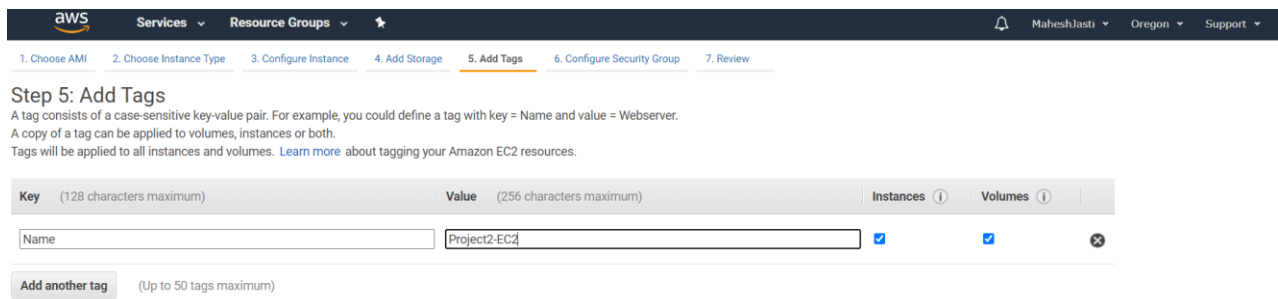
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MiB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0e54a519c999adbdb	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

Add tag..



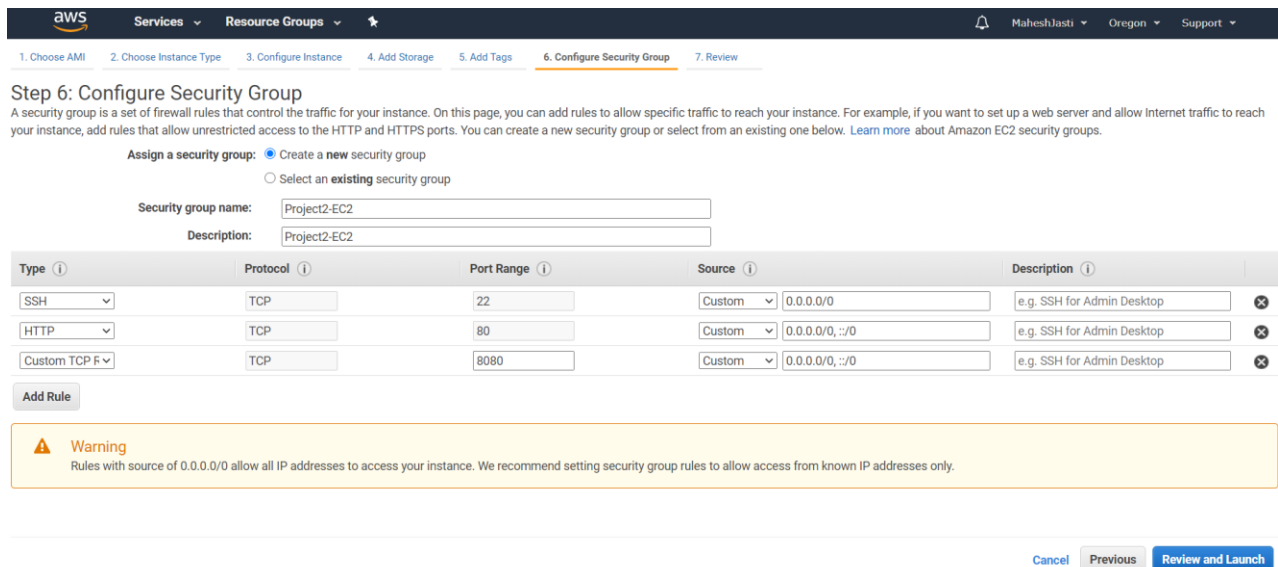
Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value	Instances	Volumes
Name	Project2-EC2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

Add the security groups as below... then click review and launch..



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a **new** security group ☐ Select an **existing** security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0:::0	e.g. SSH for Admin Desktop
Custom TCP R	TCP	8080	Custom 0.0.0.0/0:::0	e.g. SSH for Admin Desktop

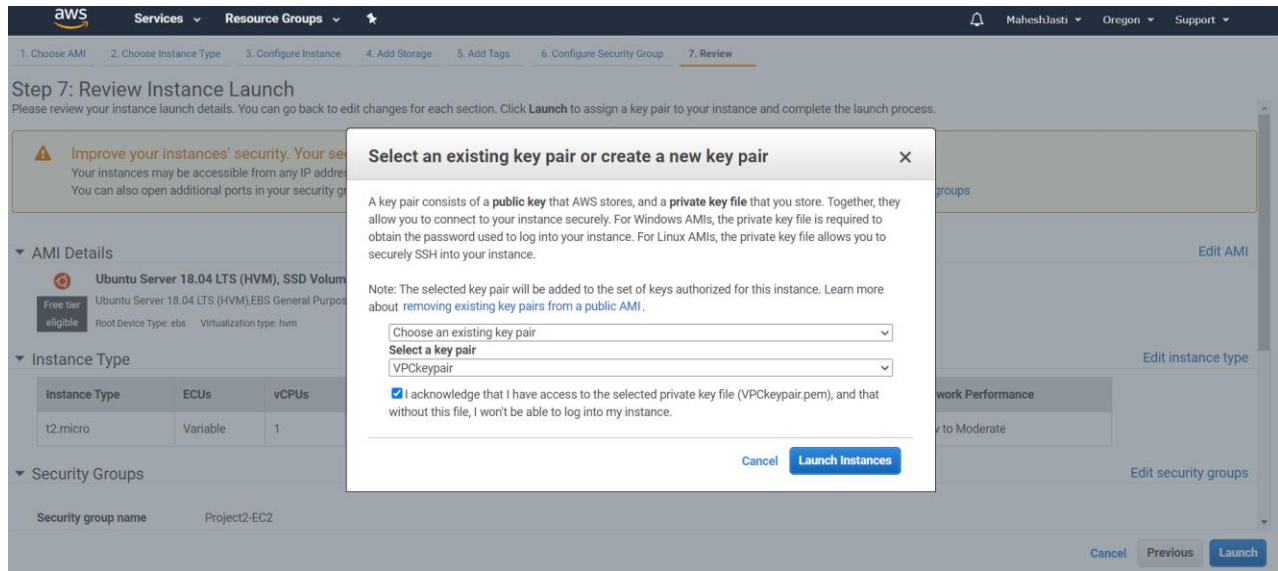
[Add Rule](#)

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

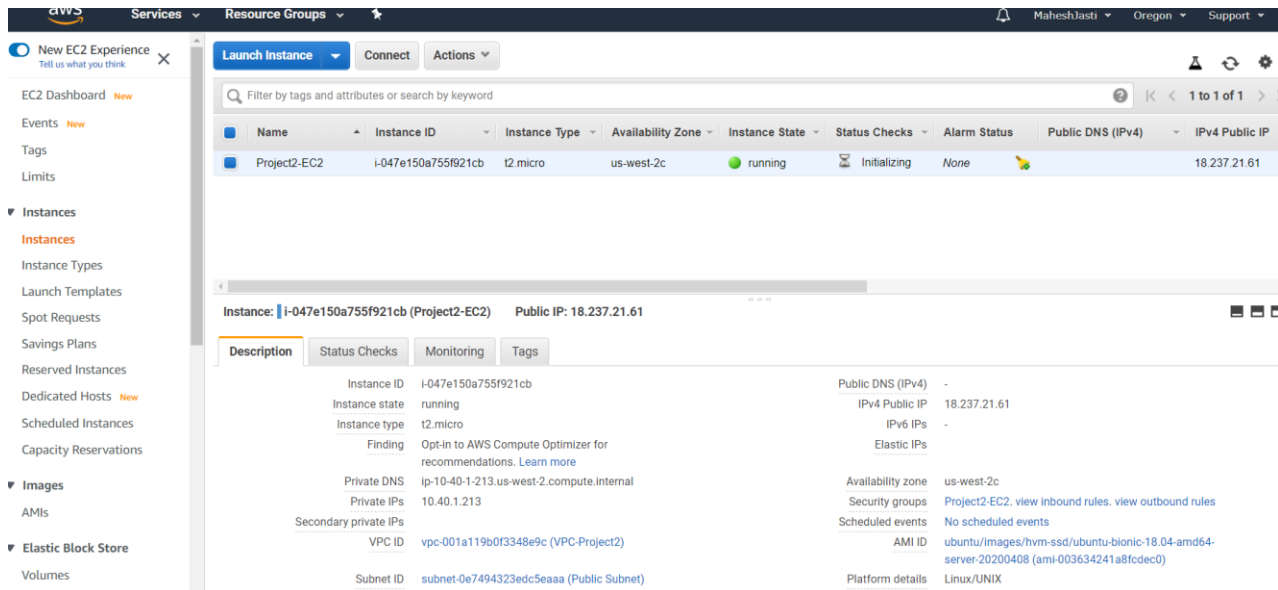
[Cancel](#) [Previous](#) [Review and Launch](#)

Select the key pair and click on "Launch Instances" button

PGPCC – Project Implementation Task



EC2 instance has been created with public IP as 18.237.21.61



SSH to EC2 Instance and copy the program into the EC2 instance..

SSH thru putty..

```

ubuntu@ip-10-40-1-213: ~
System information as of Mon Aug 10 10:26:42 UTC 2020

System load:  0.0           Processes:           86
Usage of /:   13.7% of 7.69GB Users logged in:      0
Memory usage: 15%          IP address for eth0: 10.40.1.213
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-40-1-213:~$

```

Execute the below commands..

sudo apt update

```

ubuntu@ip-10-40-1-213: ~
Get:20 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe
Translation-en [4588 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [80
8 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [25
4 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packag
es [75.6 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-
en [16.5 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages
[689 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en
[228 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packag
es [8112 B]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-
en [2852 B]
Fetched 19.1 MB in 4s (4879 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-10-40-1-213:~$

```

`sudo apt install python-pip -y`

```
ubuntu@ip-10-40-1-213: ~
Setting up python-dbus (1.2.6-1) ...
Setting up python-ipaddress (1.0.17-1) ...
Setting up python-pip (9.0.1-2.3-ubuntu1.18.04.1) ...
Setting up python-all (2.7.15-rc1-1) ...
Setting up python-setuptools (39.0.1-2) ...
Setting up gcc-7 (7.5.0-3ubuntu1~18.04) ...
Setting up g++-7 (7.5.0-3ubuntu1~18.04) ...
Setting up gcc (4:7.4.0-1ubuntu2.3) ...
Setting up python-keyrings.alt (3.0-1) ...
Setting up dpkg-dev (1.19.0.5ubuntu2.3) ...
Setting up python-all-dev (2.7.15-rc1-1) ...
Setting up python-cryptography (2.1.4-1ubuntu1.3) ...
Setting up g++ (4:7.4.0-1ubuntu2.3) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up python-secretstorage (2.3.1-2) ...
Setting up python-keyring (10.6.0-1) ...
Setting up build-essential (12.4ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
ubuntu@ip-10-40-1-213:~$
```

`python -m pip install --upgrade pip setuptools`

```
ubuntu@ip-10-40-1-213: ~
Setting up python-all-dev (2.7.15-rc1-1) ...
Setting up python-cryptography (2.1.4-1ubuntu1.3) ...
Setting up g++ (4:7.4.0-1ubuntu2.3) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up python-secretstorage (2.3.1-2) ...
Setting up python-keyring (10.6.0-1) ...
Setting up build-essential (12.4ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
ubuntu@ip-10-40-1-213:~$ python -m pip install --upgrade pip setuptools
Collecting pip
  Downloading https://files.pythonhosted.org/packages/bd/b1/56a834acdbe23b486dea16aaf4c27ed28eb292695b90d01dff96c96597de/pip-20.2.1-py2.py3-none-any.whl (1.5MB)
    100% |#####| 1.5MB 736kB/s
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/e1/b7/182161210a13158cd3cc41ee19a4def54496b74f2817cc147006ec932b4/setuptools-44.1.1-py2.py3-none-any.whl (583kB)
    100% |#####| 583kB 1.8MB/s
Installing collected packages: pip, setuptools
Successfully installed pip-20.2.1 setuptools-44.1.1
ubuntu@ip-10-40-1-213:~$
```

`sudo apt install virtualenv -y`

```
ubuntu@ip-10-40-1-213: ~
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 virtualenv all 15.1.0+ds-1.1 [4476 B]
Fetched 269 kB in 0s (9972 kB/s)
Selecting previously unselected package python3-lib2to3.
(Reading database ... 63855 files and directories currently installed.)
Preparing to unpack .../python3-lib2to3_3.6.9-1~18.04_all.deb ...
Unpacking python3-lib2to3 (3.6.9-1~18.04) ...
Selecting previously unselected package python3-distutils.
Preparing to unpack .../python3-distutils_3.6.9-1~18.04_all.deb ...
Unpacking python3-distutils (3.6.9-1~18.04) ...
Selecting previously unselected package python3-virtualenv.
Preparing to unpack .../python3-virtualenv_15.1.0+ds-1.1_all.deb ...
Unpacking python3-virtualenv (15.1.0+ds-1.1) ...
Selecting previously unselected package virtualenv.
Preparing to unpack .../virtualenv_15.1.0+ds-1.1_all.deb ...
Unpacking virtualenv (15.1.0+ds-1.1) ...
Setting up python3-lib2to3 (3.6.9-1~18.04) ...
Setting up python3-distutils (3.6.9-1~18.04) ...
Setting up python3-virtualenv (15.1.0+ds-1.1) ...
Setting up virtualenv (15.1.0+ds-1.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
ubuntu@ip-10-40-1-213:~$
```

virtualenv ~/.virtualenvs/djangodev

```

ubuntu@ip-10-40-1-213: ~
Selecting previously unselected package python3-lib2to3.
(Reading database ... 63855 files and directories currently installed.)
Preparing to unpack .../python3-lib2to3 3.6.9-1~18.04_all.deb ...
Unpacking python3-lib2to3 (3.6.9-1~18.04) ...
Selecting previously unselected package python3-distutils.
Preparing to unpack .../python3-distutils 3.6.9-1~18.04_all.deb ...
Unpacking python3-distutils (3.6.9-1~18.04) ...
Selecting previously unselected package python3-virtualenv.
Preparing to unpack .../python3-virtualenv 15.1.0+ds-1.1_all.deb ...
Unpacking python3-virtualenv (15.1.0+ds-1.1) ...
Selecting previously unselected package virtualenv.
Preparing to unpack .../virtualenv 15.1.0+ds-1.1_all.deb ...
Unpacking virtualenv (15.1.0+ds-1.1) ...
Setting up python3-lib2to3 (3.6.9-1~18.04) ...
Setting up python3-distutils (3.6.9-1~18.04) ...
Setting up python3-virtualenv (15.1.0+ds-1.1) ...
Setting up virtualenv (15.1.0+ds-1.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
ubuntu@ip-10-40-1-213:~$ virtualenv ~/.virtualenvs/djangodev
Running virtualenv with interpreter /usr/bin/python2
New python executable in /home/ubuntu/.virtualenvs/djangodev/bin/python2
Also creating executable in /home/ubuntu/.virtualenvs/djangodev/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
ubuntu@ip-10-40-1-213:~$

```

source ~/.virtualenvs/djangodev/bin/activate

```

ubuntu@ip-10-40-1-213: ~
(Reading database ... 63855 files and directories currently installed.)
Preparing to unpack .../python3-lib2to3 3.6.9-1~18.04_all.deb ...
Unpacking python3-lib2to3 (3.6.9-1~18.04) ...
Selecting previously unselected package python3-distutils.
Preparing to unpack .../python3-distutils 3.6.9-1~18.04_all.deb ...
Unpacking python3-distutils (3.6.9-1~18.04) ...
Selecting previously unselected package python3-virtualenv.
Preparing to unpack .../python3-virtualenv 15.1.0+ds-1.1_all.deb ...
Unpacking python3-virtualenv (15.1.0+ds-1.1) ...
Selecting previously unselected package virtualenv.
Preparing to unpack .../virtualenv 15.1.0+ds-1.1_all.deb ...
Unpacking virtualenv (15.1.0+ds-1.1) ...
Setting up python3-lib2to3 (3.6.9-1~18.04) ...
Setting up python3-distutils (3.6.9-1~18.04) ...
Setting up python3-virtualenv (15.1.0+ds-1.1) ...
Setting up virtualenv (15.1.0+ds-1.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
ubuntu@ip-10-40-1-213:~$ virtualenv ~/.virtualenvs/djangodev
Running virtualenv with interpreter /usr/bin/python2
New python executable in /home/ubuntu/.virtualenvs/djangodev/bin/python2
Also creating executable in /home/ubuntu/.virtualenvs/djangodev/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
ubuntu@ip-10-40-1-213:~$ source ~/.virtualenvs/djangodev/bin/activate
(djangodev) ubuntu@ip-10-40-1-213:~$

```

pip install Django

```

ubuntu@ip-10-40-1-213: ~
Setting up python3-virtualenv (15.1.0+ds-1.1) ...
Setting up virtualenv (15.1.0+ds-1.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
ubuntu@ip-10-40-1-213:~$ virtualenv ~/.virtualenvs/djangodev
Running virtualenv with interpreter /usr/bin/python2
New python executable in /home/ubuntu/.virtualenvs/djangodev/bin/python2
Also creating executable in /home/ubuntu/.virtualenvs/djangodev/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
ubuntu@ip-10-40-1-213:~$ source ~/.virtualenvs/djangodev/bin/activate
(djangodev) ubuntu@ip-10-40-1-213:~$ pip install django
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please
upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 will drop s
upport for Python 2.7 in January 2021. More details about Python 2 support in pi
p can be found at https://pip.pypa.io/en/latest/development/release-process/#pyt
hon-2-support
Collecting django
  Downloading Django-1.11.29-py2.py3-none-any.whl (6.9 MB)
    |#####| 6.9 MB 5.4 MB/s
Collecting pytz
  Downloading pytz-2020.1-py2.py3-none-any.whl (510 kB)
    |#####| 510 kB 18.0 MB/s
Installing collected packages: pytz, django
Successfully installed django-1.11.29 pytz-2020.1
(djangodev) ubuntu@ip-10-40-1-213:~$

```


pip install boto3

```

ubuntu@ip-10-40-1-213: ~
Collecting jmespath<1.0.0,>=0.7.1
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting s3transfer<0.4.0,>=0.3.0
  Downloading s3transfer-0.3.3-py3-none-any.whl (69 kB)
  | 69 kB 4.9 MB/s
Collecting docutils<0.16,>=0.10
  Downloading docutils-0.15.2-py2-none-any.whl (548 kB)
  | 548 kB 8.0 MB/s
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python-dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
  | 227 kB 11.7 MB/s
Collecting urllib3<1.26,>=1.20; python_version != "3.4"
  Downloading urllib3-1.25.10-py2.py3-none-any.whl (127 kB)
  | 127 kB 9.1 MB/s
Collecting futures<4.0.0,>=2.2.0; python_version == "2.7"
  Downloading futures-3.3.0-py2-none-any.whl (16 kB)
Collecting six>=1.5
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: jmespath, docutils, six, python-dateutil, urllib3,
botocore, futures, s3transfer, boto3
Successfully installed boto3-1.14.38 botocore-1.17.38 docutils-0.15.2 futures-3.
3.0 jmespath-0.10.0 python-dateutil-2.8.1 s3transfer-0.3.3 six-1.15.0 urllib3-1.
25.10
(djangodev) ubuntu@ip-10-40-1-213:~$

```

Move the files to Ubuntu instance. I have used the winscp..

D:\(G:\03 Managed Services)\Project-2\docproc-new\				/opt/			
Name	Size	Type	Changed	Name	Size	Changed	Rights
..		Parent directory	7/27/2020 4:34:43 PM	..		8/10/2020 3:35:19 PM	root
api		File folder	3/6/2019 11:53:18 AM	api		8/10/2020 4:10:09 PM	ubuntu
docproc		File folder	10/29/2018 9:00:48 AM	docproc		8/10/2020 4:10:13 PM	ubuntu
db.sqlite3	37 KB	SQLite3 File	3/29/2018 7:43:29 PM	db.sqlite3	37 KB	3/29/2018 7:43:29 PM	ubuntu
manage.py	1 KB	Python File	3/29/2018 7:33:48 PM	manage.py	1 KB	3/29/2018 7:33:48 PM	ubuntu

All the files has been moved to opt folder..

```

ubuntu@ip-10-40-1-213: /opt
Collecting urllib3<1.26,>=1.20; python_version != "3.4"
  Downloading urllib3-1.25.10-py2.py3-none-any.whl (127 kB)
  | 127 kB 9.1 MB/s
Collecting futures<4.0.0,>=2.2.0; python_version == "2.7"
  Downloading futures-3.3.0-py2-none-any.whl (16 kB)
Collecting six>=1.5
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: jmespath, docutils, six, python-dateutil, urllib3,
botocore, futures, s3transfer, boto3
Successfully installed boto3-1.14.38 botocore-1.17.38 docutils-0.15.2 futures-3.
3.0 jmespath-0.10.0 python-dateutil-2.8.1 s3transfer-0.3.3 six-1.15.0 urllib3-1.
25.10
(djangodev) ubuntu@ip-10-40-1-213:~$ cd /opt
(djangodev) ubuntu@ip-10-40-1-213:/opt$ ls -l
total 0
(djangodev) ubuntu@ip-10-40-1-213:/opt$ sudo chown ubuntu:ubuntu -R /opt
(djangodev) ubuntu@ip-10-40-1-213:/opt$ cd /opt
(djangodev) ubuntu@ip-10-40-1-213:/opt$ ls -l
total 52
drwxrwxr-x 3 ubuntu ubuntu 4096 Aug 10 10:40 api
-rw-rw-r-- 1 ubuntu ubuntu 37888 Mar 29 2018 db.sqlite3
drwxrwxr-x 2 ubuntu ubuntu 4096 Aug 10 10:40 docproc
-rw-rw-r-- 1 ubuntu ubuntu 805 Mar 29 2018 manage.py
(djangodev) ubuntu@ip-10-40-1-213:/opt$

```

Execute the following 2 commands

Cd docproc/

Ls -al

```

ubuntu@ip-10-40-1-213: /opt/docproc
Successfully installed boto3-1.14.38 botocore-1.17.38 docutils-0.15.2 futures-3.
3.0 jmespath-0.10.0 python-dateutil-2.8.1 s3transfer-0.3.3 six-1.15.0 urllib3-1.
25.10
(djangodev) ubuntu@ip-10-40-1-213:~$ cd /opt
(djangodev) ubuntu@ip-10-40-1-213:/opt$ ls -l
total 0
(djangodev) ubuntu@ip-10-40-1-213:/opt$ sudo chown ubuntu:ubuntu -R /opt
(djangodev) ubuntu@ip-10-40-1-213:/opt$ cd /opt
(djangodev) ubuntu@ip-10-40-1-213:/opt$ ls -l
total 52
drwxrwxr-x 3 ubuntu ubuntu 4096 Aug 10 10:40 api
-rw-rw-r-- 1 ubuntu ubuntu 37888 Mar 29 2018 db.sqlite3
drwxrwxr-x 2 ubuntu ubuntu 4096 Aug 10 10:40 docproc
-rw-rw-r-- 1 ubuntu ubuntu 805 Mar 29 2018 manage.py
(djangodev) ubuntu@ip-10-40-1-213:/opt$ cd docproc/
(djangodev) ubuntu@ip-10-40-1-213:/opt/docproc$ ls -al
total 20
drwxrwxr-x 2 ubuntu ubuntu 4096 Aug 10 10:40 .
drwxr-xr-x 4 ubuntu ubuntu 4096 Aug 10 10:40 ..
-rw-rw-r-- 1 ubuntu ubuntu 0 Mar 29 2018 __init__.py
-rw-rw-r-- 1 ubuntu ubuntu 3129 Mar 29 2018 settings.py
-rw-rw-r-- 1 ubuntu ubuntu 798 Mar 29 2018 urls.py
-rw-rw-r-- 1 ubuntu ubuntu 392 Mar 29 2018 wsgi.py
(djangodev) ubuntu@ip-10-40-1-213:/opt/docproc$

```

Update the Dynamo Region and S3 Target bucket name in views.py file..

```

ubuntu@ip-10-40-1-213: /opt/api
GNU nano 2.9.3 views.py
-- 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

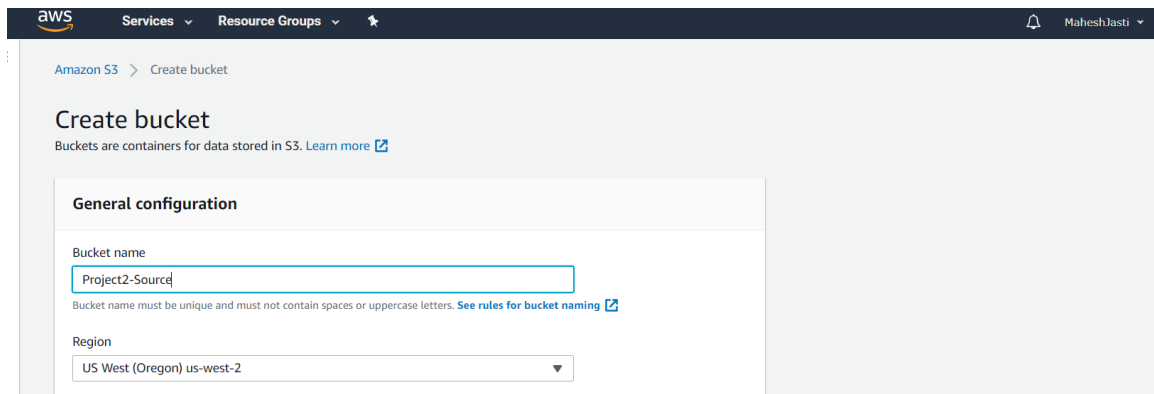
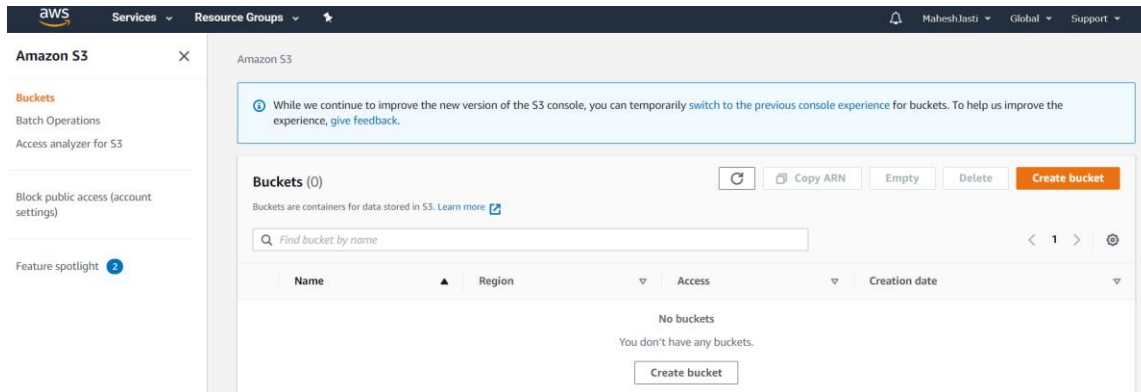
dynamodb = boto3.resource('dynamodb', region_name='us-west-2')
s3_target_bucket = 'project2-targetcsv'

[ Read 207 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^ Go To Line

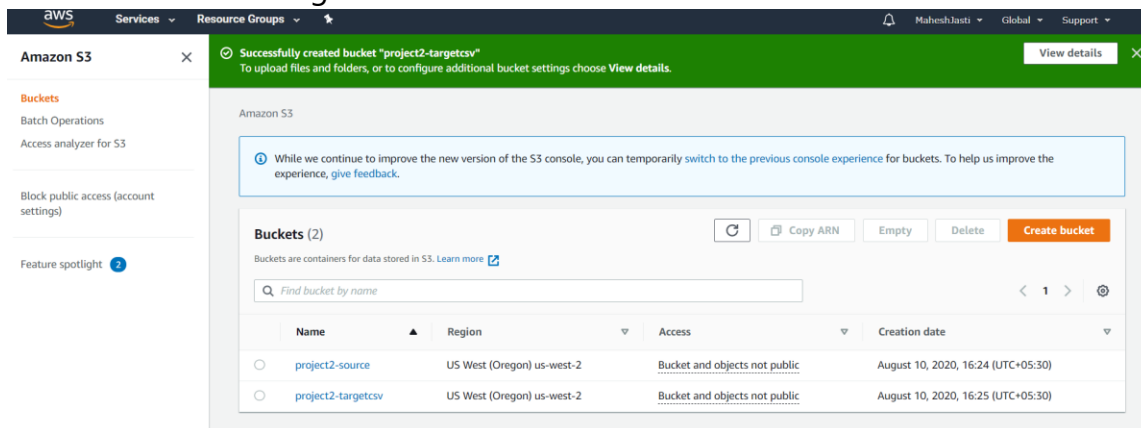
```

No create a S3 buckets for input and target..

Project2-source – Input Bucket



S3 for Source and Target



Setup SNS for S3

The screenshot shows the AWS SNS console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a user profile 'MaheshJasti'. Below the navigation bar, the 'Name' field is set to 's3upload' with a note: 'Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).' The 'Display name - optional' field is also set to 's3upload' with a note: 'To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. Info. Maximum 100 characters, including hyphens (-) and underscores (_).' The 'Permissions' section is partially visible at the bottom.

Setup access policy to “everyone”

The screenshot shows the 'Access policy - optional' section in the AWS SNS console. It states: 'This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. Info'. Under 'Choose method', the 'Basic' option is selected (Use simple criteria to define a basic access policy), while 'Advanced' is unselected (Use a JSON object to define an advanced access policy). Under 'Define who can publish messages to the topic', the 'Everyone' option is selected (Anybody can publish). Under 'Define who can subscribe to this topic', the 'Everyone' option is also selected (Any AWS account can subscribe to the topic). A 'JSON preview' section on the right shows the following JSON policy:

```
{
  "Sid": "__default_statement_ID",
  "Effect": "Allow",
  "Principal": {
    "AWS": "*"
  },
  "Action": [
    "SNS:Publish",
    "SNS:RemovePermission",
    "SNS:SetTopicAttributes",
    "SNS>DeleteTopic",
    "SNS:ListSubscriptionsByTopic",
    "SNS:GetTopicAttributes",
    "SNS:Receive",
    "SNS:AddPermission",
    "SNS:Subscribe"
  ],
}
```

Topic is created

The screenshot shows the AWS SNS console after the topic 's3upload' has been created. A green banner at the top says 'Topic s3upload created successfully. You can create subscriptions and send messages to them from this topic.' Below the banner, the breadcrumb 'Amazon SNS > Topics > s3upload' is shown. The topic name 's3upload' is displayed with 'Edit', 'Delete', and 'Publish message' buttons. The 'Details' section shows the following information:

Name	s3upload	Display name	s3upload
ARN	arn:aws:sns:us-west-2:554932232731:s3upload	Topic owner	554932232731

PGPCC – Project Implementation Task

Create a subscription..

The screenshot shows the 'Create subscription' page in the AWS Management Console. The breadcrumb navigation is 'Amazon SNS > Subscriptions > Create subscription'. The page title is 'Create subscription'. Under the 'Details' section, the 'Topic ARN' is 'arn:aws:sns:us-west-2:554932232731:s3upload'. The 'Protocol' is 'HTTP'. The 'Endpoint' is 'http://18.237.21.61/sns'. The checkbox 'Enable raw message delivery' is checked. A blue information box at the bottom states: 'After your subscription is created, you must confirm it. Info'.

The screenshot shows the 'Edit subscription' page for the subscription ARN 'd64fa486-3aee-4c58-84b0-86b5abaa90f5'. The breadcrumb navigation is 'Amazon SNS > Subscriptions > Edit d64fa486-3aee-4c58-84b0-86b5abaa90f5'. The page title is 'Edit d64fa486-3aee-4c58-84b0-86b5abaa90f5'. Under the 'Details' section, the 'Topic' is 'arn:aws:sns:us-west-2:554932232731:s3upload', the 'Protocol' is 'HTTP', and the 'Endpoint' is 'http://18.237.21.61:8080/sns'. The checkbox 'Enable raw message delivery' is unchecked. Below the details, there is a section for 'Subscription filter policy - optional' with a note: 'This policy filters the messages that a subscriber receives. Info'.

Click on the request confirmation

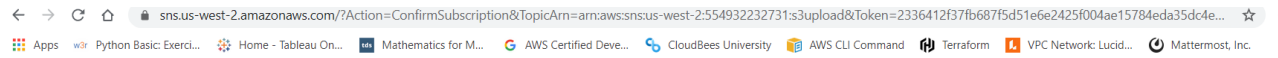
The screenshot shows the 'Subscription' page for the topic 's3upload'. The breadcrumb navigation is 'Amazon SNS > Subscriptions > s3upload'. The page title is 's3upload'. The 'Details' section shows the 'Name' as 's3upload', 'Display name' as 's3upload', 'ARN' as 'arn:aws:sns:us-west-2:554932232731:s3upload', and 'Topic owner' as '554932232731'. Below the details, there are tabs for 'Subscriptions', 'Access policy', 'Delivery retry policy (HTTP/S)', 'Delivery status logging', 'Encryption', and 'Tags'. The 'Subscriptions' tab is active, showing a table with one subscription. The table has columns for 'ID', 'Endpoint', 'Status', and 'Protocol'. The subscription is in 'Pending confirmation' status. The 'Request confirmation' button is highlighted.

ID	Endpoint	Status	Protocol
○ Pending confirmation	http://18.237.21.61/sns	○ Pending confirmation	HTTP

PGPCC – Project Implementation Task

```
ubuntu@ip-10-40-1-213:~$ curl -X POST https://sns.us-west-2.amazonaws.com/?Action=ConfirmSubscription&TopicArn=arn:aws:sns:us-west-2:554932232731:s3upload
{"Message": "You have chosen to subscribe to the topic arn:aws:sns:us-west-2:554932232731:s3upload. To confirm the subscription, visit the SubscribeURL included in this message.", "SubscribeURL": "https://sns.us-west-2.amazonaws.com/?Action=ConfirmSubscription&TopicArn=arn:aws:sns:us-west-2:554932232731:s3upload&Token=2336412f37fb687f5d51e6e2425f004ae15784eda35dc4e22e4a2a0a1d467f284ed5d5dfc62f23a28c6080019fcdcf0a316539c293a22767f6ba20f47019a316d1f703575dae3362db91901bca249a60a5670ab8fe74a63a43", "Timestamp": "2020-08-10T11:36:02.033Z", "SignatureVersion": "1", "Signature": "H0Jagm8G1YK3aCwmy4f0C4sofPWLKFFJ3AkcR6G1P0Fz9M0F3JCL28La2V+UeXp943YnPUw1M0wvYwacUUVIM0p9K0E505Fp9KtCC/HL858CW31VZAgfCfK0VFF14E43e+408Ea41Q0TFR0FEWboeP3m0B4+ZRENGh1/UA7V1wCQ6dcm/1EgKv2e2K8VTA/jaV1s170au2zy1B81qUL0J53Ma1K0GpR+HTW0WynRf9yWfqn382a2d1dCv1Pnfad/XpTs7X0K8AbJm5GfoLNtN1d9F30b3qhrvFW0Gw6uSL13nYiqct82B7Be2hEYA==", "SigningCertURL": "https://sns.us-west-2.amazonaws.com/SimpleNotificationService-a86cb10b4e1f29c941702d737128f7b6.pem"}
Request method = POST
The S3 JSON is {
  "Type": "SubscriptionConfirmation",
  "MessageId": "1ee17387-e032-4ee1-a0e9-d2610292cf83",
  "Token": "2336412f37fb687f5d51e6e2425f004ae15784eda35dc4e22e4a2a0a1d467f284ed5d5dfc62f23a28c6080019fcdcf0a316539c293a22767f6ba20f47019a316d1f703575dae3362db91901bca249a60a5670ab8fe74a63a43",
  "TopicArn": "arn:aws:sns:us-west-2:554932232731:s3upload",
  "Message": "You have chosen to subscribe to the topic arn:aws:sns:us-west-2:554932232731:s3upload. To confirm the subscription, visit the SubscribeURL included in this message.",
  "SubscribeURL": "https://sns.us-west-2.amazonaws.com/?Action=ConfirmSubscription&TopicArn=arn:aws:sns:us-west-2:554932232731:s3upload&Token=2336412f37fb687f5d51e6e2425f004ae15784eda35dc4e22e4a2a0a1d467f284ed5d5dfc62f23a28c6080019fcdcf0a316539c293a22767f6ba20f47019a316d1f703575dae3362db91901bca249a60a5670ab8fe74a63a43",
  "Timestamp": "2020-08-10T11:36:02.033Z",
  "SignatureVersion": "1",
  "Signature": "H0Jagm8G1YK3aCwmy4f0C4sofPWLKFFJ3AkcR6G1P0Fz9M0F3JCL28La2V+UeXp943YnPUw1M0wvYwacUUVIM0p9K0E505Fp9KtCC/HL858CW31VZAgfCfK0VFF14E43e+408Ea41Q0TFR0FEWboeP3m0B4+ZRENGh1/UA7V1wCQ6dcm/1EgKv2e2K8VTA/jaV1s170au2zy1B81qUL0J53Ma1K0GpR+HTW0WynRf9yWfqn382a2d1dCv1Pnfad/XpTs7X0K8AbJm5GfoLNtN1d9F30b3qhrvFW0Gw6uSL13nYiqct82B7Be2hEYA==",
  "SigningCertURL": "https://sns.us-west-2.amazonaws.com/SimpleNotificationService-a86cb10b4e1f29c941702d737128f7b6.pem"}
}
Internal Server Error: /sns
Traceback (most recent call last):
  File "/home/ubuntu/.virtualenvs/djangodev/local/lib/python2.7/site-packages/django/core/handlers/exception.py", line 41, in inner
    response = get_response(request)
  File "/home/ubuntu/.virtualenvs/djangodev/local/lib/python2.7/site-packages/django/core/handlers/base.py", line 187, in _get_response
    response = self.process_exception_by_middleware(e, request)
  File "/home/ubuntu/.virtualenvs/djangodev/local/lib/python2.7/site-packages/django/core/handlers/base.py", line 185, in _get_response
    response = wrapped_callback(request, *callback_args, **callback_kwargs)
  File "/home/ubuntu/.virtualenvs/djangodev/local/lib/python2.7/site-packages/django/views/decorators/csrf.py", line 58, in wrapped_view
    return view_func(*args, **kwargs)
  File "/opt/api/views.py", line 31, in message
    process_document(request.body)
  File "/opt/api/views.py", line 80, in process_document
    ...
```

Now copy the url and open in a new browser



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<ConfirmSubscriptionResponse xmlns="http://sns.amazonaws.com/doc/2010-03-31/">
  <ConfirmSubscriptionResult>
    <SubscriptionArn>arn:aws:sns:us-west-2:554932232731:s3upload:d64fa486-3aee-4c58-84b0-86b5abaa90f5</SubscriptionArn>
  </ConfirmSubscriptionResult>
  <ResponseMetadata>
    <RequestId>ae123505-af42-5c35-a5f8-62ae9cab489b</RequestId>
  </ResponseMetadata>
</ConfirmSubscriptionResponse>
```

Once you refresh the page your subscription is confirmed.

Amazon SNS × **s3upload** [Edit] [Delete] [Publish message]

Details

Name	Display name
s3upload	s3upload
ARN	Topic owner
arn:aws:sns:us-west-2:554932232731:s3upload	554932232731

Subscriptions | Access policy | Delivery retry policy (HTTP/S) | Delivery status logging | Encryption | Tags

Subscriptions (1) [Edit] [Delete] [Request confirmation] [Confirm subscription] **Create subscription**

Search

ID	Endpoint	Status	Protocol
d64fa486-3aee-4c58-84b0-86b5abaa90f5	http://18.237.21.61:8080/sns	Confirmed	HTTP

Setup event trigger for S3 source bucket (project2-source bucket)

Name ⓘ

s3uploadsource

Events ⓘ

☒ PUT

☐ POST

☐ COPY

☐ Multipart upload completed

☐ All object create events

☐ Object in RRS lost

☐ Permanently deleted

☐ Delete marker created

☐ All object delete events

☐ Restore initiated

☐ Restore completed

☐ Replication time missed threshold

☐ Replication time completed after threshold

☐ Replication time not tracked

☐ Replication failed

Prefix ⓘ

e.g. images/

Suffix ⓘ

e.g. .jpg

Send to ⓘ

SNS Topic

SNS

s3upload

Disabled

Now attach the EC2-Project2 role to the EC2 instance..

Instances > Attach/Replace IAM Role

Attach/Replace IAM Role

Select an IAM role to attach to your instance. If you don't have any IAM roles, choose Create new IAM role to create a role in the IAM console. If an IAM role is already attached to your instance, the IAM role you choose will replace the existing role.

Instance ID i-047e150a755f921cb (Project2-EC2) ⓘ

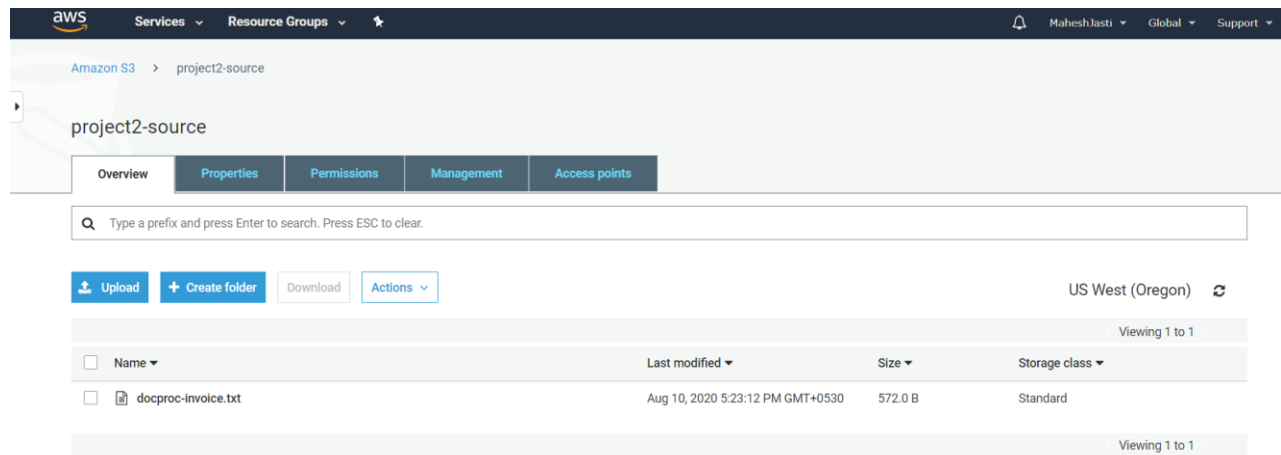
IAM role* EC2-Project2 ⓘ [Create new IAM role](#) ⓘ

* Required

[Cancel](#) [Apply](#)

Now upload a file in S3 project2-source bucket.

PGPCC – Project Implementation Task



The screenshot displays the AWS S3 console interface for a bucket named 'project2-source'. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The breadcrumb trail shows 'Amazon S3 > project2-source'. The bucket name 'project2-source' is prominently displayed. Below the bucket name, there are tabs for 'Overview', 'Properties', 'Permissions', 'Management', and 'Access points', with 'Overview' being the active tab. A search bar is present with the placeholder text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar, there are buttons for 'Upload', 'Create folder', 'Download', and 'Actions'. The region is set to 'US West (Oregon)'. A table lists the contents of the bucket, showing one file named 'docproc-invoice.txt' with a size of 572.0 B, last modified on 'Aug 10, 2020 5:23:12 PM GMT+0530', and stored in the 'Standard' storage class. The table has columns for 'Name', 'Last modified', 'Size', and 'Storage class'. The interface also shows 'Viewing 1 to 1' at the bottom right of the table.

Name	Last modified	Size	Storage class
docproc-invoice.txt	Aug 10, 2020 5:23:12 PM GMT+0530	572.0 B	Standard

The program has been processed and the data is parsed and loaded into the dynamo table..

PGPCC – Project Implementation Task

```
ubuntu@ip-10-40-1-213: /opt
bscriptionArn=arn:aws:sns:us-west-2:554932232731:s3upload:d64fa486-3aee-4c58-84b
0-86b5abaa90f5"
}
Will read the file docproc-invoice.txt from the bucket project2-source
Line-> *****
Line-> Invoice
Line-> *****
Line-> Customer-ID: sm-LK0080145
Found Customer-ID sm-LK0080145
Line-> Inv-ID: inv-00001
Found Invoice-ID inv-00001
Line-> Dated: Mar 31 2018
Line-> From: Space Mechanics Inc.
Line-> To: Starfleet Interplanatory HQ
Line-> Amount: 7650
Line-> SGST: 500
Line-> Total: 8150
Line-> InWords: Space Rupees Eight thousand one hundred fifty only
Line->
Line-> Items
Line-> 1) Matter antimatter fusion controller; 2000; 200
Line-> 2) External inertial damper; 3100; 180
Line-> 3) Type X phaser; 500; 35
Line-> 4) 2MW Fusion reactor; 1200; 85
Line-> 5) Transporter base dial; 850
Found dated Mar 31 2018
Found fromcust Space Mechanics Inc.
Found tocust Starfleet Interplanatory HQ
Found amt 7650
Found sgst 500
Found tot 8150
Found words Space Rupees Eight thousand one hundred fifty only
CSV -> sm-LK0080145,inv-00001, Mar 31 2018, Space Mechanics Inc., Starfleet Inte
rplanatory HQ, 7650, 500, 8150, Space Rupees Eight thousand one hundred fifty on
ly

*****
Creating table invoice
DONE

*****
Inserting data in the table
Done.
Written new s3 file 77772933 sm-LK0080145 inv-00001.csv
Uploading S3 object content sm-LK0080145,inv-00001, Mar 31 2018, Space Mechanics
Inc., Starfleet Interplanatory HQ, 7650, 500, 8150, Space Rupees Eight thousand
one hundred fifty only
Done
```

The items are loaded into dynamo db

PGPCC – Project Implementation Task

The screenshot shows the AWS DynamoDB console interface. On the left, the navigation pane includes 'DynamoDB', 'Tables', 'Backups', 'Reserved capacity', 'Preferences', 'DAX', 'Dashboards', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main panel displays the 'invoice' table with tabs for 'Overview', 'Items', 'Metrics', 'Alarms', 'Capacity', 'Indexes', 'Global Tables', 'Backups', 'Contributor Insights', 'Triggers', 'Access control', and 'Tags'. The 'Items' tab is active, showing a list of items. One item is selected, and the 'Edit item' dialog is open. The dialog shows the item's structure with the following details:

- Item {4}
- csvdts String: sm-LK0080145,inv-00001, Mar 31 2018, Space Mechanics Inc., Starfleet Interplanetary HQ, 7650, 500, 8150, Space Rupees Eight thousand one hundred fifty only
- cust_id String: sm-LK0080145
- details String: *****\nCustomer-ID: sm-LK0080145\nInv-ID: inv-00001\nDated: Mar 31 2018\nFrom: Space Mechanics Inc.\nTo: Starfleet Interplanetary HQ\nAmount: 7650\nSGST: 500\nTotal: 8150\nInWords: Space Rupees Eight thousand one hundred fifty only\nItems\nMatter antimatter fusion controller; 2000; 200\n2) External inertial damper; 3100; 180\n3) Type X phaser; 500; 35\n4) 2MW Fusion reactor; 1200; 85\n5) Transporter base dial; 850\n
- inv_id String: inv-00001

The 'Edit item' dialog has 'Cancel' and 'Save' buttons at the bottom right.

Go to Athena window.

The screenshot shows the AWS Athena console 'Connect data source' screen. The top navigation bar includes 'Athena', 'Query editor', 'Saved queries', 'History', 'Data sources', 'Workgroup: primary', 'Settings', 'Tutorial', 'Help', and 'What's new'. The 'Data sources' tab is active. The screen displays two steps: 'Step 1: Choose a data source' and 'Step 2: Connection details'. Under 'Step 1', there are two options: 'Query data in Amazon S3' (selected) and 'Query a data source (beta)'. Below these, there are two options for 'Choose a metadata catalog': 'AWS Glue data catalog' (selected) and 'Apache Hive metastore'. The 'Next' button is visible at the bottom right.

Add a table

The screenshot shows the AWS Athena console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a user profile 'MaheshJasti' in 'Oregon'. Below this, the 'Athena' section is active, with tabs for 'Query editor', 'Saved queries', 'History', 'Data sources', and 'Workgroup: primary'. The 'Data sources' tab is selected. The main content area is titled 'Connect data source'. It has two steps: 'Step 1: Choose a data source' and 'Step 2: Connection details'. 'Step 2' is the current step. It shows 'Connection details: AWS Glue data catalog'. Below this, it explains that Athena will connect to data in Amazon S3 using AWS Glue. There are two radio buttons: 'Set up crawler in AWS Glue to retrieve schema information automatically' (unselected) and 'Add a table and enter schema information manually' (selected). At the bottom right of the form are 'Cancel', 'Previous', and 'Continue to add table' buttons.

Enter the required details

The screenshot shows the 'Add table' screen in the AWS Athena console. The breadcrumb is 'Databases > Add table'. There are four steps: 'Step 1: Name & Location', 'Step 2: Data Format', 'Step 3: Columns', and 'Step 4: Partitions'. 'Step 1' is the current step. It contains the following fields and options:

- Database:** A dropdown menu with 'Create a new database' selected. Below it, text says: 'Choose an existing database or create a new one by selecting "Create new database".'
- Name of the new database:** A text input field containing 's3data'.
- Table Name:** A text input field containing 'sourceinvoice'. Below it, text says: 'Name of the new table. Table names must be globally unique. Table names tend to correspond to the directory where the data will be stored.'
- Location of Input Data Set:** A text input field containing 's3://project2-source/'. To its right is an unchecked checkbox for 'Encrypted data set' with an information icon.
- External:** A checked checkbox. Below it, a note says: 'Note: Amazon Athena only allows you to create tables with the EXTERNAL keyword. Dropping a table created with the External keyword does not drop the underlying data.'

At the bottom left, there is a blue 'Next' button.

PGPCC – Project Implementation Task

The screenshot shows the AWS Athena console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The main header shows 'Athena' and various tabs like 'Query editor', 'Saved queries', 'History', 'Data sources', and 'Workgroup: primary'. On the left, an 'ACTION' sidebar contains a link to 'Add table'. The main content area is titled 'Databases > Add table' and shows the 'Step 2: Data Format' tab selected. Under 'Data Format', several options are listed with radio buttons: 'Apache Web Logs', 'CSV' (which is selected), 'TSV', 'Text File with Custom Delimiters', 'JSON', 'Parquet', and 'ORC'.

Add the columns as per the input text doc..

This screenshot shows the 'Step 3: Columns' tab of the 'Add table' wizard. It displays three column definitions, each with a 'Column Name' input field and a 'Column type' dropdown menu. The first column is named 'Customer-ID' with a type of 'string'. The second column is named 'Inv-ID' with a type of 'string'. The third column is named 'Dated' with a type of 'string'. Below each column name input field, there is a note: 'Column name must be single words that start with a letter or a digit.' Below each column type dropdown, there is a note: 'Type for this column. Certain advanced types (namely, structs) are not exposed in this interface.'

Add all the other columns and click on Create Table..

PGPCC – Project Implementation Task

The screenshot shows the AWS Athena Query Editor interface. The left sidebar displays the 'Data source' as 'AwsDataCatalog' and the 'Database' as 's3data'. Under 'Tables (1)', the 'sourceinvoice' table is listed with columns: customer-id (string), inv-id (string), dated (string), from (string), to (string), amount (string), sgst (string), total (string), and inwords (string). The main query editor shows a 'New query 3' tab with a SQL query to create an external table named 'sourceinvoice'. The query is: 1 CREATE EXTERNAL TABLE IF NOT EXISTS s3data.sourceinvoice (, 2 'Customer-ID' string, 3 'Inv-ID' string, 4 'Dated' string, 5 'From' string, 6 'To' string, 7 'Amount' string, 8 'SGST' string, 9 'Total' string, 10 'Inwords' string, 11), 12 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe' 13 WITH SERDEPROPERTIES (14 'serialization.format' = ',', 15 'field.delim' = ',', 16) LOCATION 's3://project2-targetcsv/' 17 TBLPROPERTIES ('has_encrypted_data'='false'); 18 19. Below the query, there are buttons for 'Run query', 'Save as', 'Create', 'Format query', and 'Clear'. The status bar indicates '(Run time: 1.52 seconds, Data scanned: 0.15 KB)'.

Now write the below query and click on Run Query by selecting ..

Select * from sourceinvoice

The screenshot shows the AWS Athena Query Editor interface with the same setup as the previous screenshot, but with a different SQL query. The query is: 1 CREATE EXTERNAL TABLE IF NOT EXISTS s3data.sourceinvoice (, 2 'Customer-ID' string, 3 'Inv-ID' string, 4 'Dated' string, 5 'From' string, 6 'To' string, 7 'Amount' string, 8 'SGST' string, 9 'Total' string, 10 'Inwords' string, 11), 12 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe' 13 WITH SERDEPROPERTIES (14 'serialization.format' = ',', 15 'field.delim' = ',', 16) LOCATION 's3://project2-targetcsv/' 17 TBLPROPERTIES ('has_encrypted_data'='false'); 18 19 select * from sourceinvoice; Below the query, there are buttons for 'Run query', 'Save as', 'Create', 'Format query', and 'Clear'. The status bar indicates '(Run time: 1.48 seconds, Data scanned: 0.15 KB)'. The 'Results' section at the bottom shows a table with 9 columns: customer-id, inv-id, dated, from, to, amount, sgst, total, and inwords. The first row of data is: 1 sm-LK0080145 inv-00001 Mar 31 2018 Space Mechanics Inc. Starfleet Interplanetary HQ 7650 500 8150 Space Rupees Eight thousand one hundred fifty only.

Section 6: Lessons learnt / Observations

1. After creating VPC, we need to create the IGWY and setup route table
2. If we need public IP, we need to select the auto assign IP address
3. Set up SNS topic with HTTP endpoint
4. A custom TCP security rule should be added for 8080 port (web server)
5. While creating table in Athena, we should store the query in some s3 location otherwise we get the error as "No output location provided. An output location is required either through the Workgroup result configuration setting or as an API input. (Service: AmazonAthena; Status Code: 400; Error Code: InvalidRequestException)"
6. Remember to use 8080 while check in browser
http://<public-ip>:8080/sns
7. While creating subscription, we have to give http://<public-ip>:8080/sns