



AWS Project
Team: TITANS
Harshith Raj
Vinayak Pote
Yathish Kumar
Nitish Kumar Singh

INDEX

Part-1

Sl No.	Topic	Pg No.
1.	Creating VPC-1	4
2.	Creating subnets	4-5
3.	Creating Internet Gateway	5
4.	Creating Route Table	6-7
5.	Creating NAT Gateway	8
6.	Creating Private Route	8-9
7.	Creating Security Groups	9
8.	Creating Instances	10
9.	Connecting to EC2	11
10.	Testing EC2	11
11.	Creating AMI	12
12.	Creating Template	12
13.	Creating AutoScaling Group	12-14
14.	Creating RDS	15
15.	Creating S3 bucket	16
16.	Creating DynamoDB table	16
17.	Data Getting uploaded on DynamoDB table	17-18
18.	Creating role for S3	19
19.	Mounting S3 bucket on EC2 instance	19-20
20.	Output	20

Part-2

A. Connecting S3 to Athena using Glue		
Sl No.	Topic	Pg No.
1.	Creating buckets	21
2.	Creating Crawlers	22
3.	Creating workgroup in Athena	22
4.	Selecting database and running query	23
5.	Output	23
B. Connecting S3 to Redshift using Glue		
1.	Creating VPC Peering	24
2.	Creating EFS	24-25
3.	Mounting EFS on instances.	26
4.	Creating S3 bucket with dataset	26
5.	Creating Redshift cluster	27
6.	Making the connection for query builder	27
7.	Creating 1 st crawler for S3 bucket	28
8.	Creating connection for Redshift	28-29
9.	Output	30

Part-1

1. Creating VPC-1

The screenshot shows the AWS VPC Management Console. A success message at the top says "You successfully created **vpc-0c798878832badda3 / titan-vpc-1**". The main pane displays the details of the newly created VPC, including its ID, state, and network settings. The VPC ID is **vpc-0c798878832badda3**, State is **Available**, and the IPv4 CIDR is **192.168.126.0/24**. The VPC is associated with a Main route table (**rtb-071fe6ba19ac157f1**) and a Main network ACL (**acl-0c02cc2e686f0317e**). The Details tab is selected, showing other configuration parameters like DNS resolution and Network mapping unit metrics.

- Created VPC-1 with name “titan-vpc-1”
- IP address for VPC: 192.168.126.0

2. Creating subnets

The screenshot shows the AWS Subnets Management Console. It lists four subnets under the VPC **vpc-0c798878832badda3 | titan-vpc-1**. The subnets are named **titan-vpc_1-subnet-1** through **titan-vpc_1-subnet-4**, each with a unique Subnet ID and assigned to the same VPC and IPv4 CIDR range of **192.168.126.0/24**.

Name	Subnet ID	VPC	IPv4 CIDR
titan-vpc_1-subnet-4	subnet-03a9ffe3370475d51	vpc-0c798878832badda3 titan-vpc-1	192.168.126.0/24
titan-vpc_1-subnet-3	subnet-06ed70d18df31cdf4	vpc-0c798878832badda3 titan-vpc-1	192.168.126.0/24
titan-vpc_1-subnet-2	subnet-090822fc3367d977e	vpc-0c798878832badda3 titan-vpc-1	192.168.126.0/24
titan-vpc_1-subnet-1	subnet-017d732b8c257cd3e	vpc-0c798878832badda3 titan-vpc-1	192.168.126.0/24

- Created 4 subnets connected to “titan-vpc-1”.
- 2 Public and 2 private subnets with ids below:

The screenshot shows a browser window with three tabs: "VPC Management Console", "ip calculator - Search", and "IP Subnet Calculator". The "IP Subnet Calculator" tab is active, displaying results for a subnet of 192.168.126.0/26. It shows the following table:

Network Address	Usable Host Range	Broadcast Address:
192.168.126.0	192.168.126.1 - 192.168.126.62	192.168.126.63
192.168.126.64	192.168.126.65 - 192.168.126.126	192.168.126.127
192.168.126.128	192.168.126.129 - 192.168.126.190	192.168.126.191
192.168.126.192	192.168.126.193 - 192.168.126.254	192.168.126.255

Below the table is a form for calculating subnets:

Network Class	<input type="radio"/> Any <input type="radio"/> A <input type="radio"/> B <input checked="" type="radio"/> C
Subnet	255.255.255.192 /26
IP Address	192.168.126.0
<input type="button" value="Calculate"/> <input type="button" value="Clear"/>	

3. Creating Internet Gateway

The screenshot shows the AWS VPC Management Console. The left sidebar is expanded to show the "Virtual private cloud" section, specifically the "Internet gateways" subsection. A message at the top indicates that an Internet Gateway has been created:

The following internet gateway was created: igw-04c24de771fcb0eaf - titan-IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

The main pane displays the details of the newly created Internet Gateway:

Internet gateway ID	igw-04c24de771fcb0eaf	State	Detached	VPC ID	-	Owner	024111598068
---------------------	-----------------------	-------	----------	--------	---	-------	--------------

Below the table is a "Tags" section:

Key	Value
Name	titan-IGW

- Created Internet Gateway with name “titan-IGW”.
- Attached it to the created VPC “titan-vpc-1”.

4. Creating Route Table

The screenshot shows the AWS VPC Management Console. In the top navigation bar, there are tabs for 'VPC Management Console', 'ip calculator - Search', 'IP Subnet Calculator', and a search bar. Below the navigation bar, the AWS logo and 'Services' link are visible. The main content area displays a success message: 'Route table rtb-0b15c1d91de370495 | titan-route-public was created successfully.' The URL in the address bar is <https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0b15c1d91de370495>. The page title is 'rtb-0b15c1d91de370495 / titan-route-public'. A sub-header says 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button. On the left sidebar, under 'Route tables', the 'Route tables' section is selected, showing details like Route table ID (rtb-0b15c1d91de370495), Main (No), VPC (vpc-0c798878832badda3 | titan-vpc-1), and Owner ID (024111598068). The bottom of the screen shows the Windows taskbar with various pinned icons.

- Created Route table “titan-route-public”.
- Attached the routed table to the internet gateway “titan-IGW”.

The screenshot shows the AWS VPC Management Console. The interface is similar to the previous one, with the 'Route tables' section selected in the sidebar. A success message 'Updated routes for rtb-0b15c1d91de370495 / titan-route-public successfully' is displayed. The 'Routes' tab is active, showing two routes in the 'Routes (2)' table:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-04c24de771fc0eaf	Active	No
192.168.126.0/24	local	Active	No

The bottom of the screen shows the Windows taskbar.

- Added subnet to the Internet Gateway.

The screenshot shows the AWS VPC Management Console with the URL <https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#RouteTableDetails:RouteTable=titan-route-public>. The main content area displays a success message: "You have successfully updated subnet associations for rtb-0b15c1d91de370495 / titan-route-public." Below this, the "Subnet associations" tab is selected, showing two explicit subnet associations:

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-017d732b8c257cd3e / titan-vpc_1-subnet-1	192.168.126.0/26	-
subnet-090822fc3367d977e / titan-vpc_1-subnet-2	192.168.126.64/26	-

Below this section, there is a heading "Subnets without explicit associations (2)" with a note: "The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:".

5. Creating NAT Gateway

The screenshot shows the AWS VPC Management Console with the URL <https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#NatGatewayDetails:natGateway=titan-NAT>. The main content area displays the details of the NAT gateway "nat-0dc4433a50c8d2e09 / titan-NAT".

Details		Info	
NAT gateway ID	nat-0dc4433a50c8d2e09	Connectivity type	Public
NAT gateway ARN	arn:aws:ec2:us-east-1:024111598068:natgateway/nat-0dc4433a50c8d2e09	Elastic IP address	54.156.153.17
VPC	vpc-0c798878832badda3 / titan-vpc-1	Subnet	subnet-017d732b8c257cd3e / titan-vpc_1-subnet-1
		Created	Thursday, October 6, 2022 at 11:52:06 GMT+5:30
		Deleted	-

- Created NAT Gateway with name “titan-NAT”.

6. Creating Private Route

The screenshot shows the AWS VPC Management Console with a success message: "Route table rtb-07b8cb83feab757b9 | titan-route-private was created successfully." The details panel shows the route table ID, VPC, and owner ID. A button to "Run Reachability Analyzer" is also present.

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-07b8cb83feab757b9	No	-	-

Details Info

VPC	Owner ID
vpc-0c798878832badda3 titan-vpc-1	024111598068

- Created Private route with name “titan-private-route”.
- Added subnet to private route.

The screenshot shows the AWS VPC Management Console with a success message: "You have successfully updated subnet associations for rtb-07b8cb83feab757b9 / titan-route-private." The "Subnet associations" tab is selected, showing two explicit subnet associations:

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-06ed70d18df31cdf4 / titan-vpc_1-subnet-3	192.168.126.128/26	-
subnet-03a9ffe3370475d51 / titan-vpc_1-subnet-4	192.168.126.192/26	-

Subnets without explicit associations (0)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Adding NAT Gateway to Private Route.

The screenshot shows the AWS VPC Management Console. On the left, there's a sidebar with navigation links like 'Virtual private cloud', 'Your VPCs', 'Subnets', and 'Route tables'. The 'Route tables' section is currently selected. In the main area, a table titled 'Routes (2)' lists two entries:

Destination	Target	Status	Propagated
0.0.0.0/0	nat-0dc4433a50c8d2e09	Active	No
192.168.126.0/24	local	Active	No

7. Creating Security Group

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with sections like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Limits', and 'Instances'. The 'Instances' section is expanded, showing 'Instances New', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances New', 'Dedicated Hosts', and 'Scheduled Instances'. In the main area, it shows the details for a security group named 'sg-0d3dc0ab4d5ccada6 - titan-security_group':

Security group name	Security group ID	Description	VPC ID
titan-security_group	sg-0d3dc0ab4d5ccada6	titan-security_group-VPC-1	vpc-0c798878832badda3

Below the details, there are tabs for 'Inbound rules', 'Outbound rules', and 'Tags'. A message at the bottom says 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button.

- Created security group “titan-security-group”.

8. Creating Instances

The screenshot shows the AWS EC2 Management Console interface. On the left, a sidebar lists various services like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. Under Instances, there are sub-options for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Scheduled Instances. The main content area displays a table titled 'Instances (4)'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. Four instances are listed: 'Titan-EC2-2' (i-0196db20c22050223), 'Titan-EC2-4' (i-05fc13b449c5a9c65), 'Titan-EC2-1' (i-0dcdfbfc1c1de57f3), and 'Titan-EC2-3' (i-0c31ebddc055528e4). All instances are shown as 'Running' with a green checkmark. The 'Status check' column indicates '2/2 checks passed' for instance 1. The 'Alarm status' column shows 'No alarms' for all instances. At the bottom of the main content area, a modal window titled 'Select an instance' is open. The bottom right corner of the screen shows a Windows taskbar with icons for File Explorer, Task View, Start, and others, along with system status information.

- Created 4 instances and connected them with respective subnets.

9. Connecting to EC2

The screenshot shows a Windows terminal window with a black background and white text. The session starts with a root prompt on a Microsoft Windows 10 machine. It then shows the user attempting to SSH into an EC2 instance at 3.83.80.99, where they are prompted to accept a new host key fingerprint. The user responds with 'yes'. The terminal then shows the user navigating to the root directory and updating the package list. Finally, the user installs the Apache web server using the yum package manager. The terminal window includes a Windows taskbar at the bottom with icons for File Explorer, Task View, Start, and others, along with system status information.

```
root@ip-192-168-126-17:/home/ec2-user
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pote_vinayak@lilly.com\Downloads>ssh -i "titan-key.pem" ec2-user@3.83.80.99
The authenticity of host '3.83.80.99 (3.83.80.99)' can't be established.
EDSA key fingerprint is SHA256:Ctw/ohSy6ZijNFrMnH2J7GSmXAHMmZLd0aDhty6Q.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.83.80.99' (EDSA) to the list of known hosts.

 _|_ / 
  | \_ |_ / Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-126-17 ~]$ sudo su
[root@ip-192-168-126-17 ec2-user]# ls
[root@ip-192-168-126-17 ec2-user]# ls -l
total 0
[root@ip-192-168-126-17 ec2-user]# yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
[root@ip-192-168-126-17 ec2-user]# yum install apache
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No package apache available.
Error: Nothing to do
[root@ip-192-168-126-17 ec2-user]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-htpd for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
```

10. Testing EC2

This screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is titled "Test Page". The content of the page is as follows:

If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:
You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



11. Creating Image

This screenshot shows the AWS EC2 Management Console. The left sidebar is expanded to show the "Images" section, specifically the "AMIs" tab. A new AMI has been created, with the ID `ami-01517d96b56c77737`. The "Image summary" for this AMI is displayed in the main content area:

AMI ID	Image type	Platform details	Root device type
ami-01517d96b56c77737	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
titan-image	024111598068	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
/dev/xvda	Available	024111598068/titan-image	hvm
Boot mode	State reason	Creation date	Kernel ID
-	-	Thu Oct 06 2022 12:33:05 GMT+0530 (India Standard Time)	-
Block devices	Description	Product codes	RAM disk ID
/dev/xvda=snap-08ab1e1bb41bbb5b1:8:true:gp2	-	-	-
Deprecation time	Last launched time		
-	-		



- Created Image

12.Creating Template

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays the 'Launch template details' for 'titan-template'. The launch template ID is lt-073dba6906e8f5ab6, the name is titan-template, the default version is 1, and the owner is arn:aws:sts::024111598068:assumed-role/aws_gis_learning_env_learner/I038269. Below this, there are tabs for 'Details', 'Versions', and 'Template tags'. A separate section titled 'Launch template version details' is also visible.

- Created Template with name “titan-template”.

13.Creating Autoscaling Group

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays the 'Auto Scaling groups' details for 'titan-auto-scaling'. The group has a desired capacity of 1, a minimum capacity of 1, and a maximum capacity of 1. The auto scaling group name is titan-auto-scaling, and it was created on Thu Oct 06 2022 12:56:27 GMT+0530 (India Standard Time). The ARN is arn:aws:autoscaling:us-east-1:024111598068:autoScalingGroup:449e0fe2-3599-4b13-a328-bba414219703:autoScalingGroupName/titan-auto-scaling. Below this, there are tabs for 'Details', 'Activity', 'Automatic scaling', 'Instance management', 'Monitoring', and 'Instance refresh'.

- Created autoscaling group “titan-auto-scaling”.
- Details of the auto-scaling group.

The screenshot shows the AWS EC2 Auto Scaling Group Details page. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances), and a Feedback section. The main content area has tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The Details tab is selected, showing 'Group details' with fields for Desired capacity (2), Minimum capacity (1), Maximum capacity (5), Auto Scaling group name (titan-A-S), Date created (Thu Oct 06 2022 16:17:02 GMT+0530 (India Standard Time)), and Amazon Resource Name (ARN) (arn:aws:autoscaling:us-east-1:024111598068:autoScalingGroup:5e10b67e-7b43-4549-93ff-e06176c12e3f:autoScalingGroupName=titan-A-S). Below this is a 'Launch template' section with an 'Edit' button. At the bottom, there's a search bar, a taskbar with icons, and a footer with copyright information and date (© 2022, Amazon Web Services, Inc. or its affiliates. 06-10-2022).

```
root@ip-192-168-126-21:/home/ec2-user
top - 10:58:19 up 10 min, 1 user, load average: 1.00, 0.83, 0.45
Tasks: 97 total, 3 running, 52 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99.0 us, 1.0 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 988932 total, 412248 free, 75684 used, 561000 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 774600 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
3283 root 20 0 114648 776 708 R 99.9 0.1 7:57.99 yes
 1 root 20 0 123500 5380 3920 S 0.0 0.5 0:02.04 systemd
 2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
 3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
 4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
 6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-ev
 8 root 0 -20 0 0 0 I 0.0 0.0 0:00.04 kworker/0:1H-ev
 9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
10 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_tasks_rude_
11 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_tasks_trace
12 root 20 0 0 0 0 S 0.0 0.0 0:00.02 ksoftirqd/0
13 root 20 0 0 0 0 I 0.0 0.0 0:00.07 rcu_sched
14 root rt 0 0 0 0 S 0.0 0.0 0:00.00 migration/0
16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
18 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs
19 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
20 root 20 0 0 0 0 I 0.0 0.0 0:00.05 kworker/u30:1-x
22 root 20 0 0 0 0 S 0.0 0.0 0:00.02 kauditd
264 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd
265 root 20 0 0 0 0 S 0.0 0.0 0:00.00 oom_reaper
266 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 writeback
268 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kcompactd0
269 root 25 5 0 0 0 S 0.0 0.0 0:00.00 ksmd
270 root 39 19 0 0 0 S 0.0 0.0 0:00.00 khugepaged
325 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kintegrityd
327 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kblockd
328 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 blkcg_punt_bio
680 root 20 0 0 0 0 S 0.0 0.0 0:00.00 xen-balloon
686 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 tpm_dev_wq
692 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 md
695 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 edac-poller
700 root -51 0 0 0 0 S 0.0 0.0 0:00.00 watchdogd
798 root 20 0 0 0 0 I 0.0 0.0 0:00.02 kworker/0:2-xfs
849 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kswapd0
```

The screenshot shows the AWS EC2 Instances page. The top navigation bar includes links for RDS Management, Target groups, Auto Scaling, Modify template, and EC2 Management. The search bar contains "auto scaling in". The region is set to N. Virginia. The user is identified as "aws_gis_learning_env_learner" with session ID "0241-1159-8068".

The left sidebar has sections for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances), and Feedback.

The main content area has tabs for Details, Activity, Automatic scaling, **Instance management**, Monitoring, and Instance refresh. The **Instance management** tab is selected. It displays two instances:

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/
i-05676a5484d5238bb	InService	t2.micro	-	titan-template V
i-07ccfa66a47d25124	InService	t2.micro	-	titan-template V

Below the instances, there is a section for Lifecycle hooks (0) with a "Create lifecycle hook" button.

The bottom status bar shows the Windows taskbar with various pinned icons and the date/time as 06-10-2022 16:28.

This screenshot is identical to the one above, showing the AWS EC2 Instances page with the same layout, navigation, and instance details. The main difference is the number of instances listed in the "Instances" table:

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/
i-049a23d6fe826a5d2	Pending	t2.micro	-	titan-template V
i-05676a5484d5238bb	InService	t2.micro	-	titan-template V
i-07ccfa66a47d25124	InService	t2.micro	-	titan-template V
i-086c674e15746dca4	Pending	t2.micro	-	titan-template V

14.Creating RDS

The screenshot shows the AWS RDS Management console. The left sidebar has 'Amazon RDS' selected. Under 'Databases', 'titan-db' is listed. The main panel shows the 'Summary' tab for the 'titan-db' database. Key details include:

DB identifier	CPU	Status	Class
titan-db	2.16%	Available	db.t3.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MariaDB	us-east-1c

Below the summary, there are tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. The 'Connectivity & security' tab is active, showing sections for 'Endpoint & port', 'Networking', and 'Security'. At the bottom of the page, there's a feedback link and a search bar.

- Created RDS database “titan-db”.
- Connected RDS to a private instance.

```
root@ip-192-168-126-134:/home/ec2-user
Total download size: 8.8 M
Installed size: 49 M
Is this ok [y/d/N]: y
Downloading packages:
mariadb-5.5.68-1.amzn2.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.x86_64
  Verifying   : 1:mariadb-5.5.68-1.amzn2.x86_64
                                         | 8.8 MB  00:00:00
                                         1/1
                                         1/1

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2

Complete!
[root@ip-192-168-126-134 ec2-user]# mysql -h titan-db.comu5rhzxjbr.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 81
Server version: 10.6.8-MariaDB-log managed by https://aws.amazon.com/rds/
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

15.Creating S3 bucket

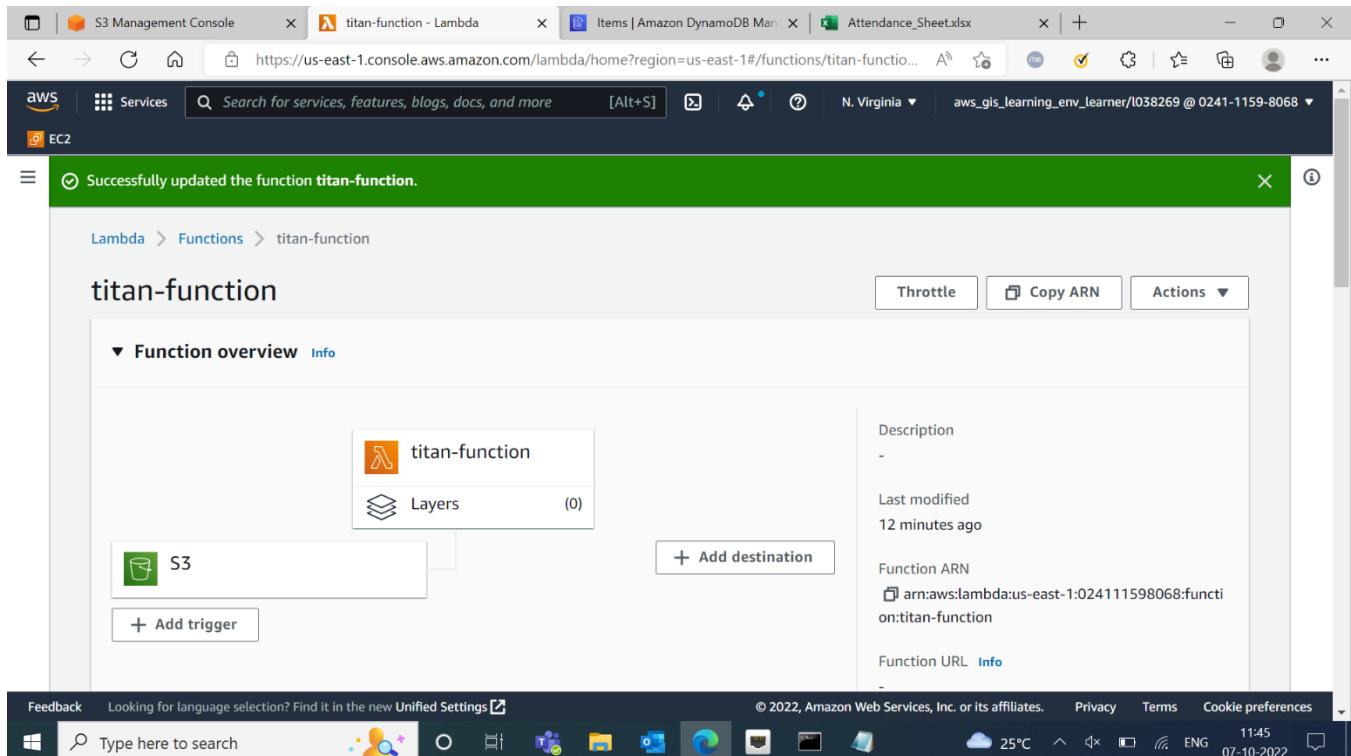
The screenshot shows the AWS S3 console interface. On the left, a sidebar titled 'Amazon S3' lists various options like 'Buckets', 'Access Points', and 'Storage Lens'. The main area is titled 'titanbucket' and shows the 'Objects' tab selected. A sub-section titled 'Objects (0)' is displayed, with instructions about objects in Amazon S3. Below this are buttons for 'Create folder' and 'Upload', and a search bar for 'Find objects by prefix'. A table at the bottom shows columns for Name, Type, Last modified, Size, and Storage class, with a note 'No objects'.

- Created S3 bucket with name “titanbucket”.

16.Creating DynamoDB table

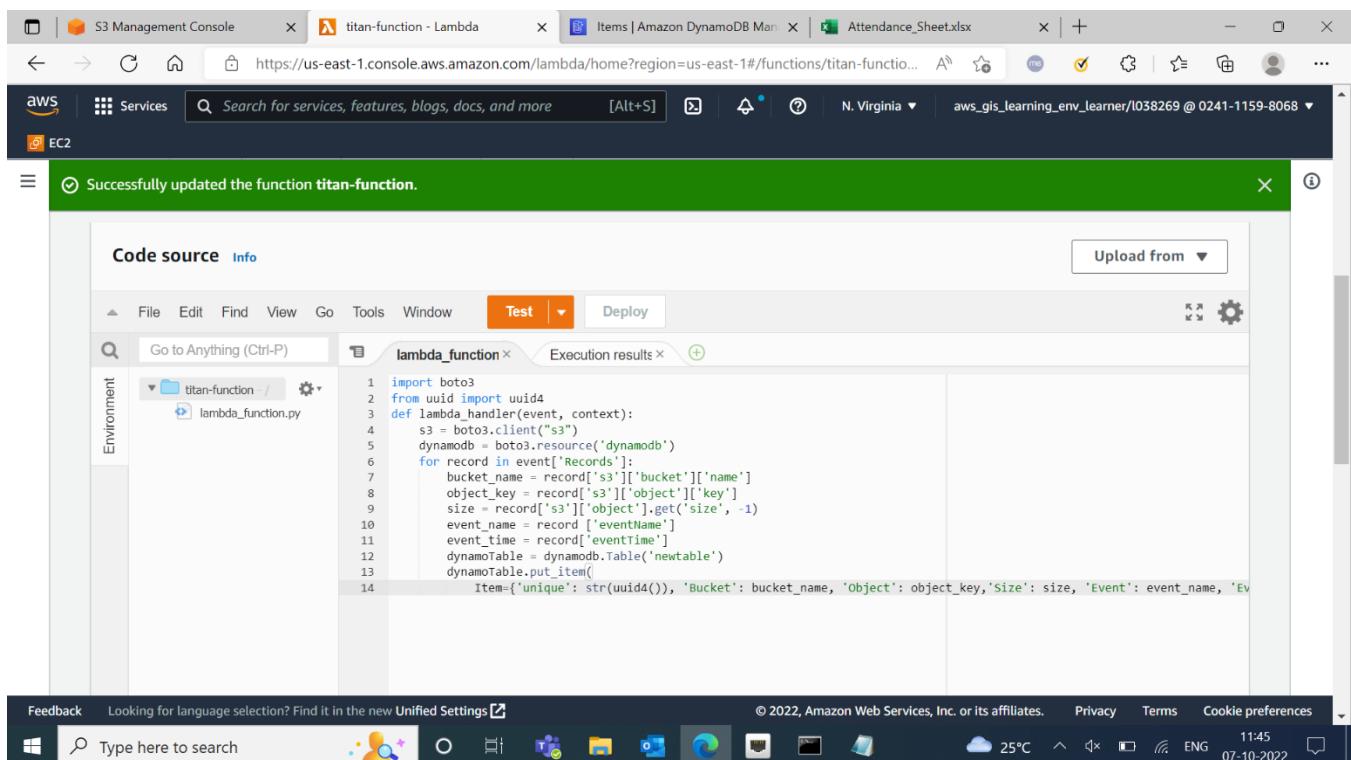
The screenshot shows the AWS DynamoDB console interface. On the left, a sidebar titled 'DynamoDB' lists options like 'Tables', 'Update settings', and 'Explore items'. The main area is titled 'newtable' and shows the 'Overview' tab selected. It displays general information such as 'Partition key unique (String)', 'Sort key -', 'Capacity mode Provisioned', and 'Table status Active'. There is also a section for 'Additional info'.

17. Uploading data in DynamoDB



The screenshot shows the AWS Lambda console. A green success message at the top says "Successfully updated the function titan-function." Below it, the "Function overview" section is visible. It shows the function name "titan-function" and its ARN. There are tabs for "Code source" and "Info". On the left, there's a trigger configuration panel with "S3" selected as the trigger type. On the right, there are fields for "Description", "Last modified", and "Function URL". The status bar at the bottom indicates the function was last modified 12 minutes ago.

- Added code to the lambda function.
- Uploaded data in the table.



The screenshot shows the AWS Lambda console with the "Code source" tab selected. The code editor displays a Python script named "lambda_function.py". The script imports boto3 and defines a lambda_handler function that processes S3 records to insert data into a DynamoDB table. The code includes logic to extract bucket and object names from the S3 event, calculate object size, and create a DynamoDB item with unique, bucket, object, size, and Event fields. The status bar at the bottom indicates the code was last modified 12 minutes ago.

```
1 import boto3
2 from uuid import uuid4
3 def lambda_handler(event, context):
4     s3 = boto3.client("s3")
5     dynamodb = boto3.resource('dynamodb')
6     for record in event['Records']:
7         bucket_name = record['s3']['bucket']['name']
8         object_key = record['s3']['object']['key']
9         size = record['s3']['object'].get('size', -1)
10        event_name = record['eventName']
11        event_time = record['eventTime']
12        dynamoTable = dynamodb.Table('newtable')
13        dynamoTable.put_item(
14            Item={'unique': str(uuid4()), 'Bucket': bucket_name, 'Object': object_key, 'Size': size, 'Event': event_name, 'Ev'}
```

Screenshot of the AWS DynamoDB Management Console. The left sidebar shows 'Explore items' selected. The main area displays a table titled 'Attendance_Sheet.xlsx' with 4 items returned. The table has columns: unique, Bucket, Event, and Value. Data rows include:

unique	Bucket	Event	Value
e34ab4bb-8f9c-48fa...	example-bucket	ObjectCreate...	19
7d9ff783-b927-4dfa...	titanbucket	ObjectCreate...	20
58e4223f-e296-474e...	titanbucket	ObjectCreate...	20
2b76d5f6-7e36-4410...	example-bucket	ObjectCreate...	19

18. Creating role for S3

Screenshot of the AWS IAM Management Console. The left sidebar shows 'Roles' selected. The main area displays a role named 'titan-function'. The 'Permissions' tab is selected, showing one policy attached: 'AmazonS3FullAccess'. The 'Trust relationships', 'Tags', and 'Access Advisor' tabs are also visible.

Permissions policies (1)
You can attach up to 10 managed policies.

Policy name	Type	Description
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Management ...

- Created role for the S3 bucket with the access policy.

19. Mounting S3 bucket on EC2 instance

```
[root@ip-192-168-126-134 ~]# yum install automake fuse fuse-devel gcc-c++ git libcurl-devel libxml2-devel make openssl-devel
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package git-2.37.1-1.amzn2.0.1.x86_64 already installed and latest version
Package 1:make-3.82-24.amzn2.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
--> Package automake.noarch 0:1.13.4-3.1.amzn2 will be installed
--> Processing Dependency: autoconf >= 2.65 for package: automake-1.13.4-3.1.amzn2.noarch
--> Processing Dependency: perl(Thread::Queue) for package: automake-1.13.4-3.1.amzn2.noarch
--> Processing Dependency: perl(TAP::Parser) for package: automake-1.13.4-3.1.amzn2.noarch
--> Package fuse.x86_64 0:2.9.2-11.amzn2 will be installed
--> Package fuse-devel.x86_64 0:2.9.2-11.amzn2 will be installed
--> Package gcc-c++.x86_64 0:7.3.1-15.amzn2 will be installed
--> Processing Dependency: gcc(x86-64) = 7.3.1-15.amzn2 for package: gcc-c++-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libmpfr.so.4()(64bit) for package: gcc-c++-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libmpc.so.3()(64bit) for package: gcc-c++-7.3.1-15.amzn2.x86_64
--> Package libcurl-devel.x86_64 0:7.79.1-4.amzn2.0.1 will be installed
--> Package libxml2-devel.x86_64 0:2.9.1-6.amzn2.5.5 will be installed
--> Processing Dependency: zlib-devel for package: libxml2-devel-2.9.1-6.amzn2.5.5.x86_64
--> Processing Dependency: xz-devel for package: libxml2-devel-2.9.1-6.amzn2.5.5.x86_64
--> Package openssl-devel.x86_64 1:1.0.2k-24.amzn2.0.4 will be installed
--> Processing Dependency: krb5-devel(x86-64) for package: 1:openssl-devel-1.0.2k-24.amzn2.0.4.x86_64
--> Running transaction check
--> Package autoconf.noarch 0:2.69-11.amzn2 will be installed
--> Processing Dependency: m4 >= 1.4.14 for package: autoconf-2.69-11.amzn2.noarch
--> Processing Dependency: perl(Data::Dumper) for package: autoconf-2.69-11.amzn2.noarch
--> Package gcc.x86_64 0:7.3.1-15.amzn2 will be installed
--> Processing Dependency: cpp = 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libsanitizer >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libquadmath >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libmpx >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libitm >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libcilkrtts >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: libatomic >= 7.3.1-15.amzn2 for package: gcc-7.3.1-15.amzn2.x86_64
--> Processing Dependency: glibc-devel >= 2.2.90-12 for package: gcc-7.3.1-15.amzn2.x86_64
--> Package krb5-devel.x86_64 0:1.15.1-37.amzn2.2.4 will be installed
--> Processing Dependency: libkadm5(x86-64) = 1.15.1-37.amzn2.2.4 for package: krb5-devel-1.15.1-37.amzn2.2.4.x86_64
```



- Mounted the created S3 bucket on the EC2 instance.

```
[root@ip-192-168-126-134 ~]# make
make all-recursive
make[1]: Entering directory '/home/ec2-user/s3fs-fuse'
Making all in src
make[2]: Entering directory '/home/ec2-user/s3fs-fuse/src'
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT s3fs.o -MD -MP -MF .deps/s3fs.Tpo -o s3fs.o s3fs.cpp
mv -f .deps/s3fs.Tpo .deps/s3fs.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT s3fs_global.o -MD -MP -MF .deps/s3fs_global.Tpo -c -o s3fs_global.o s3fs_global.cpp
mv -f .deps/s3fs_global.Tpo .deps/s3fs_global.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT s3fs_help.o -MD -MP -MF .deps/s3fs_help.Tpo -c -o s3fs_help.o s3fs_help.cpp
mv -f .deps/s3fs_help.Tpo .deps/s3fs_help.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT s3fs_logger.o -MD -MP -MF .deps/s3fs_logger.Tpo -c -o s3fs_logger.o s3fs_logger.cpp
mv -f .deps/s3fs_logger.Tpo .deps/s3fs_logger.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT s3fs_xml.o -MD -MP -MF .deps/s3fs_xml.Tpo -c -o s3fs_xml.o s3fs_xml.cpp
mv -f .deps/s3fs_xml.Tpo .deps/s3fs_xml.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT metaheader.o -MD -MP -MF .deps/metaheader.Tpo -c -o metaheader.o metaheader.cpp
mv -f .deps/metaheader.Tpo .deps/metaheader.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT mpu_util.o -MD -MP -MF .deps/mpu_util.Tpo -c -o mpu_util.o mpu_util.cpp
mv -f .deps/mpu_util.Tpo .deps/mpu_util.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT mvnode.o -MD -MP -MF .deps/mvnode.Tpo -c -o mvnode.o mvnode.cpp
mv -f .deps/mvnode.Tpo .deps/mvnode.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT curl.o -MD -MP -MF .deps/curl.Tpo -c -o curl.o curl.cpp
mv -f .deps/curl.Tpo .deps/curl.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT curl_handlerpool.o -MD -MP -MF .deps(curl_handlerpool.Tpo -c -o curl_handlerpool.o curl_handlerpool.cpp
mv -f .deps(curl_handlerpool.Tpo .deps(curl_handlerpool.Po
g++ -DHAVE_CONFIG_H -I.. -I.. -D_FILE_OFFSET_BITS=64 -I/usr/include/fuse -I/usr/include/libxml2      -g -O2 -Wall -fno-exceptions -D_FILE_OFFSET_BITS=64 -D_F
ORTIFY_SOURCE=2 -MT curl_multi.o -MD -MP -MF .deps(curl_multi.Tpo -c -o curl_multi.o curl_multi.cpp
ORTIFY_SOURCE=2 -MT curl_multi.Tpo .deps(curl_multi.Po
```



```
root@ip-192-168-126-134:~# cd s3fs-fuse
[root@ip-192-168-126-134 ec2-user]# ./autogen.sh
--- Make commit hash file -----
--- Finished commit hash file ---
--- Start autotools -----
configure.ac:26: installing './config.guess'
configure.ac:26: installing './config.sub'
configure.ac:27: installing './install-sh'
configure.ac:27: installing './missing'
src/Makefile.am: installing './depcomp'
parallel-tests: installing './test-driver'
--- Finished autotools -----
[root@ip-192-168-126-134 s3fs-fuse]# ./configure --prefix=/usr --with-openssl
configure: WARNING: you should use --build, --host, --target
configure: WARNING: invalid host type: -
configure: WARNING: you should use --build, --host, --target
configure: WARNING: invalid host type: -
configure: WARNING: invalid host type: -
configure: WARNING: you should use --build, --host, --target
checking build system type... Invalid configuration `-' machine `-' not recognized
configure: error: /bin/sh ./config.sub - failed
[root@ip-192-168-126-134 s3fs-fuse]# ./configure --prefix=/usr --with-openssl
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk...
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for g++... g++
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C++ compiler... yes
```

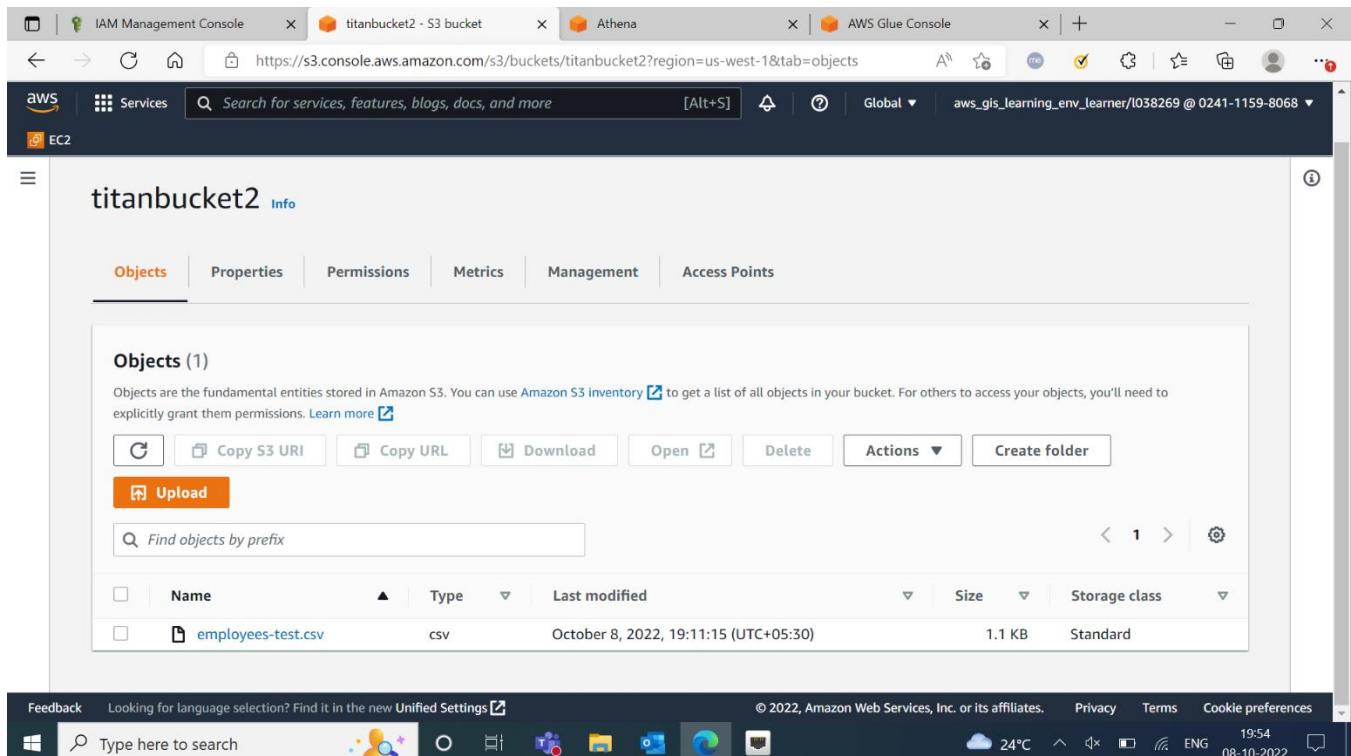
20.Output

```
root@ip-192-168-126-134:~# s3fs -o iam_role="titan-role-2" titanbucket /var/s3fs-demos
[root@ip-192-168-126-134 s3fs-fuse]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M    0  474M  0% /dev
tmpfs          483M    0  483M  0% /dev/shm
tmpfs          483M  508K  483M  1% /run
tmpfs          483M    0  483M  0% /sys/fs/cgroup
/dev/xvda1     8.0G  1.9G  6.2G  24% /
tmpfs          97M    0   97M  0% /run/user/1000
s3fs           16E    0   16E  0% /var/s3fs-demos
[root@ip-192-168-126-134 s3fs-fuse]# ls -l /var/s3fs-demos
total 6
-rw-r----- 1 root root 3778 Oct  7 06:12 employees.csv
-rw-r----- 1 root root 1710 Oct  7 06:13 Lambda.txt
[root@ip-192-168-126-134 s3fs-fuse]#
```

- Final output showing the name of the files :
 - a. Employees.csv
 - b. Lambda.txt

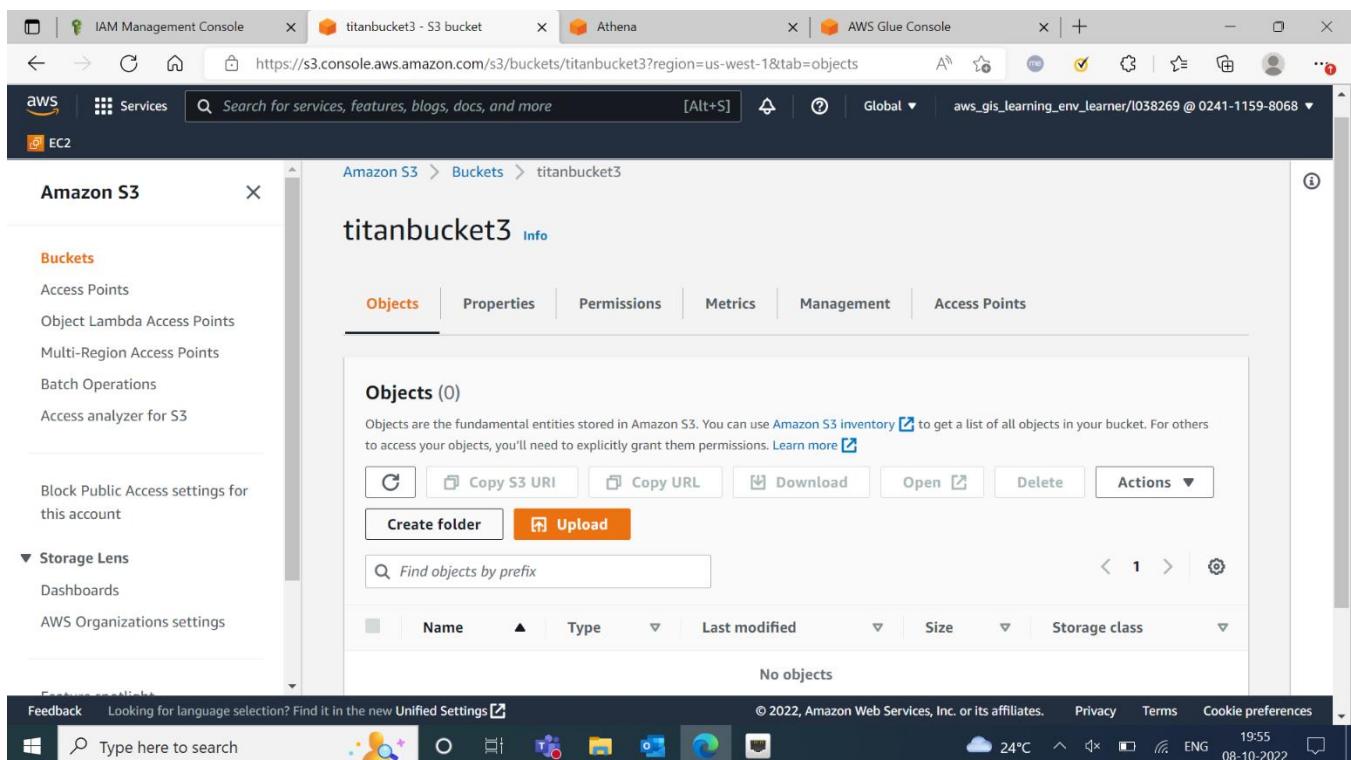
Part 2: A. Connecting S3 to Athena using glue

1. Creating buckets



The screenshot shows the AWS S3 console interface. At the top, there are tabs for IAM Management Console, titanbucket2 - S3 bucket, Athena, and AWS Glue Console. Below the tabs, the URL is https://s3.console.aws.amazon.com/s3/buckets/titanbucket2?region=us-west-1&tab=objects. The main content area is titled 'titanbucket2' and shows 'Objects (1)'. There is a table with one row, showing 'employees-test.csv' as a CSV file last modified on October 8, 2022, at 19:11:15 (UTC+05:30), with a size of 1.1 KB and a storage class of Standard. Below the table are buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, and Create folder. A search bar says 'Find objects by prefix' and a navigation bar shows page 1 of 1. The bottom of the screen shows the Windows taskbar with various icons and the date/time as 08-10-2022.

- Created a bucket with csv.



The screenshot shows the AWS S3 console interface. At the top, there are tabs for IAM Management Console, titanbucket3 - S3 bucket, Athena, and AWS Glue Console. Below the tabs, the URL is https://s3.console.aws.amazon.com/s3/buckets/titanbucket3?region=us-west-1&tab=objects. The main content area is titled 'titanbucket3' and shows 'Objects (0)'. There is a table with no rows, indicating 'No objects'. Below the table are buttons for Create folder and Upload. A search bar says 'Find objects by prefix' and a navigation bar shows page 1 of 1. The bottom of the screen shows the Windows taskbar with various icons and the date/time as 08-10-2022.

- Created second bucket to store output.

2. Creating a Crawler

The screenshot shows the AWS Glue console interface. On the left, a sidebar lists various services: Data Catalog, Databases (New), Tables (New), Stream schema registries, Schemas, Connections (New), Crawlers (New), Classifiers (New), Catalog settings, Data Integration and ETL, AWS Glue Studio, Jobs (New), Interactive Sessions, Notebooks (New), and Feedback. The main area is titled "Crawlers" and contains a message: "New console experience for AWS Crawlers available! We've redesigned the AWS Crawlers console to make it easier to use. Switch to the new console." Below this is a table showing one crawler named "titan-crawler". The table has columns for Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. The crawler "titan-crawler" is listed with a status of "Ready", logs link, last runtime of "51 secs", median runtime of "51 secs", 0 updated tables, and 1 added table. At the bottom of the page, there are links for Feedback, Unified Settings, and a copyright notice: "© 2022, Amazon Web Services, Inc. or its affiliates." followed by Privacy, Terms, and Cookie preferences. The system status bar at the bottom right shows "24°C", "ENG", and the date "08-10-2022".

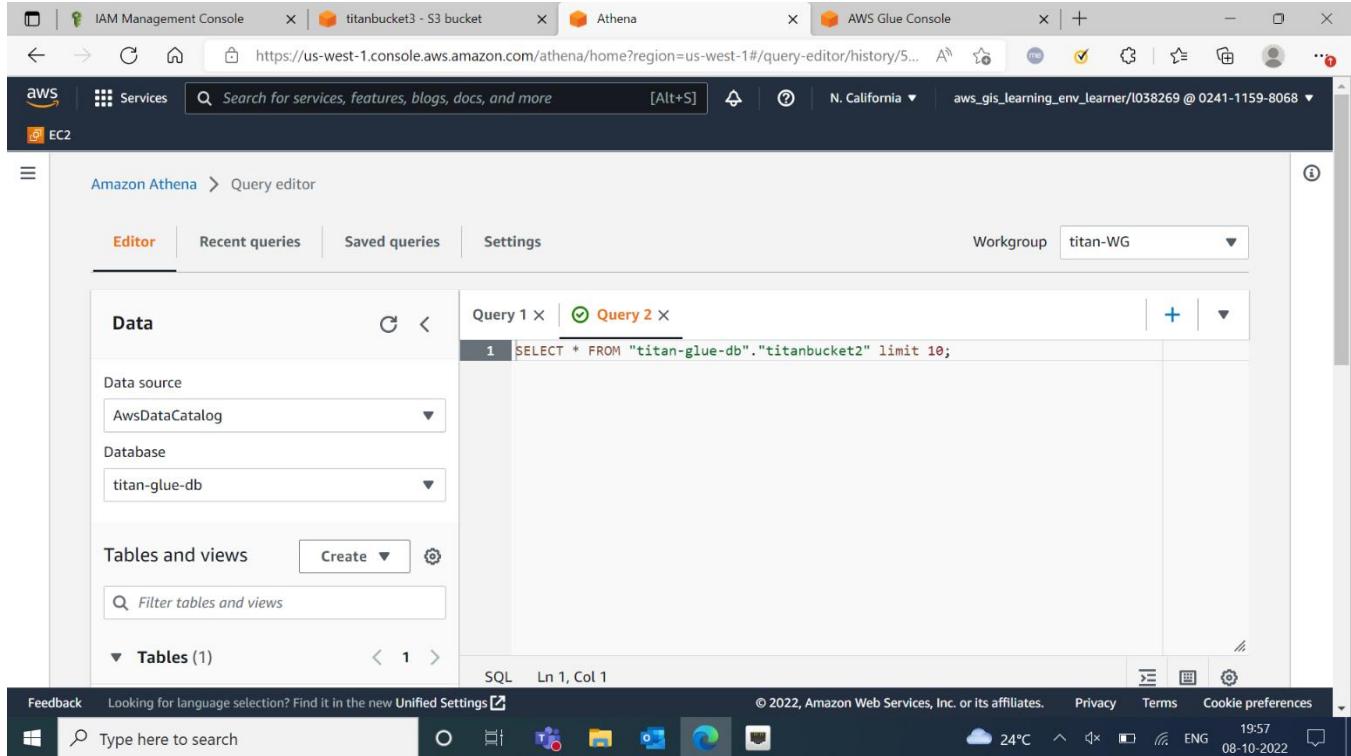
- Created a crawler named “titan-crawler”.

3. Creating workgroup in Athena

The screenshot shows the Amazon Athena console interface. On the left, a sidebar lists: Query editor, Workgroups (highlighted in orange), Data sources, and Jobs (with sub-options: Workflows, Powered by Step Functions). A "Turn on compact mode" button is also present. The main area is titled "titan-WG" and shows the "Overview details" of the workgroup. The table includes fields such as Workgroup name (titan-WG), Query engine version status (Automatic), Query result location (s3://titanbucket3/), Description (-), Override client side settings (Turned off), Encrypt query results (-), Created on (2022-10-08T19:43:50.557+05:30), Queries with requester pays buckets (Turned off), Expected bucket owner (-), Query engine version (Athena engine version 2), Workgroup ARN (arn:aws:athena:us-west-2:123456789012:workgroup/titan-WG), and Assign bucket owner full control over query results (-). At the bottom of the page, there are links for Feedback, Unified Settings, and a copyright notice: "© 2022, Amazon Web Services, Inc. or its affiliates." followed by Privacy, Terms, and Cookie preferences. The system status bar at the bottom right shows "24°C", "ENG", and the date "08-10-2022".

- Created a workgroup in Athena with name “titan-WG”.

4. Selecting Database and running Query



The screenshot shows the AWS Management Console with the Athena service selected. In the top navigation bar, the URL is https://us-west-1.console.aws.amazon.com/athena/home?region=us-west-1#/query-editor/history/5... and the region is set to N. California. The user is identified as aws_gis_learning_env_learner/I038269 @ 0241-1159-8068.

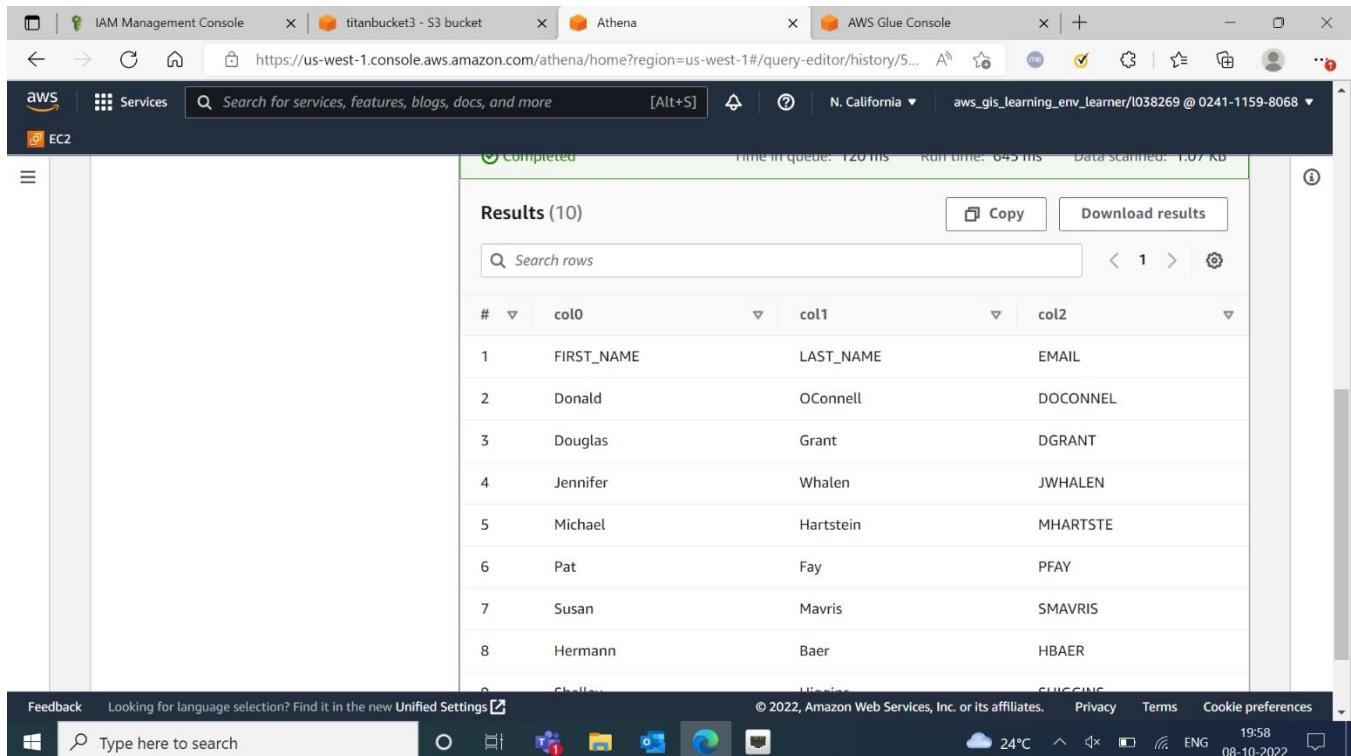
The main interface is the Query editor. The left sidebar shows the Data source (AwsDataCatalog) and Database (titan-glue-db). The right pane displays two queries:

- Query 1**: SELECT * FROM "titan-glue-db"."titanbucket2" limit 10;
- Query 2** (highlighted in green):

The status bar at the bottom indicates: Feedback, Looking for language selection? Find it in the new Unified Settings, © 2022, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences. The system tray shows the date as 08-10-2022 and the time as 19:57.

- Selected database “titan-glue-db” from “titanbucket2” and ran the query.

5. Output



The screenshot shows the AWS Management Console with the Athena service selected. The URL is https://us-west-1.console.aws.amazon.com/athena/home?region=us-west-1#/query-editor/history/5... and the region is set to N. California. The user is identified as aws_gis_learning_env_learner/I038269 @ 0241-1159-8068.

The main interface is the Results page for the completed query. The results are displayed in a table:

#	col0	col1	col2
1	FIRST_NAME	LAST_NAME	EMAIL
2	Donald	OConnell	DOCONNEL
3	Douglas	Grant	DGRANT
4	Jennifer	Whalen	JWHALEN
5	Michael	Hartstein	MHARTSTE
6	Pat	Fay	PFAY
7	Susan	Mavris	SMAVRIS
8	Hermann	Baer	HBAER

The status bar at the bottom indicates: Feedback, Looking for language selection? Find it in the new Unified Settings, © 2022, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences. The system tray shows the date as 08-10-2022 and the time as 19:58.

- Result showing entries in the selected database.

Part 2: B. Connecting S3 to Redshift using Glue

1. Creating VPC Peering

The screenshot shows the AWS VPC Peering connections page. A specific peering connection named "pcx-0d754816a5f00050e" is displayed. The "Details" tab is selected, showing the following information:

Requester owner ID	Acceptor owner ID	VPC Peering connection ARN
024111598068	024111598068	arn:aws:ec2:us-east-1:024111598068:vpc-peering-connection/pcx-0d754816a5f00050e
Peering connection ID	Requester VPC	Acceptor VPC
pcx-0d754816a5f00050e	vpc-0e8bcd8b30b563af8	vpc-0c798878832badda3 / titan-vpc-1
Status	Requester CIDRs	Acceptor CIDRs
Active	192.168.127.0/24	192.168.126.0/24
Expiration time	Requester Region	Acceptor Region
-	N. California (us-west-1)	N. Virginia (us-east-1)

The left sidebar shows navigation options like Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The "Peering connections" section is currently selected.

- Established a peering connection.

2. Creating EFS

The screenshot shows the AWS Amazon EFS File systems page. A new file system named "titan-EFS (fs-0ee12911078b211a8)" is displayed. The "General" tab is selected, showing the following configuration:

Performance mode	Automatic backups
General Purpose	Disabled
Throughput mode	Encrypted
Bursting	8e66699c-7b0b-4b40-b65f-fc6af89a0d32 (aws/elasticfilesystem)
Lifecycle management	File system state
Transition into IA: 30 days since last access	Available
Transition out of IA: On first access	
Availability zone	DNS name
Standard	fs-0ee12911078b211a8.efs.us-east-1.amazonaws.com

The left sidebar shows navigation options like File systems, Access points, AWS Backup, AWS DataSync, AWS Transfer, and Documentation. The "File systems" section is currently selected.

- Created EFS with name “titan-EFS”.
- Accessing Private EC2 using Public EC2.

```
[root@ip-192-168-126-31 ec2-user]# cd /tmp
[root@ip-192-168-126-31 tmp]# ssh -i "titan-key-R2.pem" ec2-user@192.168.127.14
Warning: Identity file titan-key-R2.pem not accessible: No such file or directory.
^C
[root@ip-192-168-126-31 tmp]# ssh -i "titan-key.pem" ec2-user@192.168.127.14

^C
[root@ip-192-168-126-31 tmp]# ssh -i "titan-key.pem" ec2-user@192.168.126.134
The authenticity of host '192.168.126.134' (192.168.126.134) can't be established.
ECDSA key fingerprint is SHA256:T05hthpRsCo6CHbrp3zZOXH14j87q4c1TgwKZYXzbgE.
ECDSA key fingerprint is MD5:23:9f:cb:69:56:ed:9b:bb:be:b9:30:45:44:a7:78:82.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.126.134' (ECDSA) to the list of known hosts.
Last login: Fri Oct  7 07:03:54 2022 from 192.168.126.31

      _\   _/_
     _\|_ /  Amazon Linux 2 AMI
    _\|_\_|_|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-126-134 ~]$
```

i-0f32728002d64f6d6 (titan-EC2-1)

PublicIPs: 54.166.16.198 PrivateIPs: 192.168.126.31

- Installing amazon-efs-utils.

```
Select root@ip-192-168-126-190:/home/ec2-user
      _\   _/_
     _\|_ /  Amazon Linux 2 AMI
    _\|_\_|_|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-126-190 ~]$ sudo su
[root@ip-192-168-126-190 ec2-user]# yum install -y amazon-efs-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package amazon-efs-utils.noarch 0:1.33.3-1.amzn2 will be installed
--> Processing Dependency: stunnel >= 4.56 for package: amazon-efs-utils-1.33.3-1.amzn2.noarch
--> Running transaction check
--> Package stunnel.x86_64 0:4.56-6.amzn2.0.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Version          Repository        Size           Arch
=====
amazon-efs-utils noarch          1.33.3-1.amzn2 amzn2-core      52 k
=====
Installing:
stunnel          x86_64          4.56-6.amzn2.0.3 amzn2-core      149 k
=====
Transaction Summary
=====
=====
Total download size: 201 k
Installed size: 508 k
Downloading packages:
(1/2): amazon-efs-utils-1.33.3-1.amzn2.noarch.rpm | 52 kB  00:00:00
(2/2): stunnel-4.56-6.amzn2.0.3.x86_64.rpm       | 149 kB  00:00:00
=====
Total
1.4 MB/s | 201 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

3. Mounting EFS on instances

The screenshot shows two terminal windows side-by-side. The left window is titled 'ec2-user@ip-172-31-15-142:~/efs' and displays a command-line session where a user navigates through directories and creates a file named 'hello.txt'. The right window is titled 'ec2-user@ip-172-31-14-41:~/efs' and shows the same file 'hello.txt' listed in the directory. This demonstrates that both instances have mounted the same EFS volume and can access the same files.

- After mounting EFS on 2 instances, we accessed same file on both the instances.

4. Creating S3 bucket with dataset

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with navigation links like 'Buckets', 'Access Points', 'Object Lambda Access Points', etc. The main area is titled 'Objects (1)' and shows a single object named 'employees-test.csv'. The object details are: Type: csv, Last modified: October 8, 2022, 19:11:15 (UTC+05:30), Size: 1.1 KB, Storage class: Standard. Below the object list are buttons for 'Create folder', 'Upload', and 'Actions'. At the bottom, there's a table with columns: Name, Type, Last modified, Size, Storage class. The single row corresponds to the 'employees-test.csv' file.

- Created a S3 bucket with dataset “employees-test.csv”.

5. Creating Redshift cluster

The screenshot shows the AWS Redshift Cluster Details page. At the top, there is a banner for 'Amazon Redshift query editor v2 is now available'. Below the banner, the cluster identifier is listed as 'titan-cluster-1'. The status is 'Available'. The node type is 'dc2.large'. The endpoint is 'titan-cluster-1.chsgjcll8gjg...'. The date created is 'October 09, 2022, 12:31 (UTC+05:30)'. The number of nodes is '1'. Storage used is '0.22% (0.35 of 160 GB used)'. The JDBC URL is 'jdbc:redshift://titan-cluste...'. The ODBC URL is 'Driver={Amazon Redshift (...}'.

- Created Redshift cluster “titan-cluster-1”.

6. Making Connection for query builder

The screenshot shows the AWS Redshift Query Editor. The connection status is 'Connected'. The database selected is 'dev'. The schema selected is 'public'. The current query tab is 'Query 1'. The interface includes a sidebar with options like 'Editor', 'Query history', 'Saved queries', and 'Scheduled queries'. The main area shows a list of tables: 'category', 'date', and 'employee'. The 'employee' table has a 'first_name' column.

7. Creating 1st crawler for S3 bucket

The screenshot shows the AWS Glue Console interface. On the left, there's a sidebar with navigation links like Data Catalog, Databases, Stream schema registries, Connections, Crawlers, Data Integration and ETL, AWS Glue Studio, Jobs, Interactive Sessions, and Notebooks. The main area is titled "titan-crawler" and displays "Crawler properties". The properties shown are:

Name	IAM role	Database	State
titan-crawler	titan-glue-role	titan-glue-db	READY

Below this, there are sections for "Description", "Security configuration", "Lake Formation configuration", and "Table prefix", each with a minus sign indicating they are collapsed. At the bottom, there's a "Maximum table threshold" field with a minus sign and a "Advanced settings" link.

- Created 1st crawler for S3 bucket “titan-crawler”.

8. Creating connection for Redshift

The screenshot shows the AWS Glue Studio interface. On the left, there's a sidebar with navigation links like Services, AWS Glue Studio, and a search bar. The main area is titled "redshift_conn" and displays "Connection details". The details shown are:

Connector type	Connection URL
JDBC	jdbc:redshift://titan-cluster-1.chsgjcll8gjg.us-west-1.redshift.amazonaws.com:5439/dev

Below this, there are sections for "Username", "Subnet", "Description", and "Last modified", each with a minus sign indicating they are collapsed. To the right of the connection URL, there are sections for "Require SSL connection" (set to false), "Security groups" (listing two security groups: sg-099e531a3a58f86f3 and sg-0756075ae14ef47a0), and "Created on" (set to 2022-10-09 12:42:24.703000). At the bottom, there's a "Class name" field.

- Created connection for redshift.

```

1 import sys
2 from awsglue.transforms import *
3 from awsglue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from awsglue.context import GlueContext
6 from awsglue.job import Job
7
8 ## @params: [TempDir, JOB_NAME]
9 args = getResolvedOptions(sys.argv, ['TempDir','JOB_NAME'])
10
11 sc = SparkContext()
12 glueContext = GlueContext(sc)
13 spark = glueContext.spark_session
14 job = Job(glueContext)

```

Python Ln 1, Col 1 Errors: 0 Warnings: 0

- Created a job for redshift.

AWS Glue

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Last updated: October 9, 2022 at 09:53:11 (UTC)

Name	Database	Location	Classification	Deprecated	View data
dev_public_employee	titan-glue-db	dev.public.employee	redshift	-	Table data
titanbucket2	titan-glue-db	s3://titanbucket2/	csv	-	Table data

- After running the 2 crawlers 2 tables were created.

9. Output

The screenshot shows the AWS Redshift Query Editor interface. On the left, there's a sidebar with navigation links: Provisioned clusters dashboard, Clusters (with Reserved nodes and Snapshots), Query editor (with Query editor v2 and Queries and loads), Datasources, and Configurations. The main area displays a table with three columns: first_name, last_name, and email. The data is as follows:

first_name	last_name	email
LAST_NAME	EMAIL	FIRST_NAME
OConnell	DOCONNEL	Donald
Grant	DGRANT	Douglas
Whalen	JWHALEN	Jennifer
Hartstein	MHARTSTE	Michael
Fay	PFAY	Pat
Mavris	SMAVRIS	Susan
Baer	HBAER	Hermann
Higgins	SHIGGINS	Shelley
Gietz	WGIETZ	William

At the bottom of the browser window, the address bar shows the URL: https://us-west-1.console.aws.amazon.com/redshiftv2/home?region=us-west-1#query-editor. The status bar indicates the session ID: aws_gis_learning_env_learner/I038269 @ 0241-1159-8068.

- Data imported into Redshift using Glue.