

AWS DAY 11 & 12 ASSIGNMENT

Question 1:

Task 1:Working with SNS

Ss1:SNS console

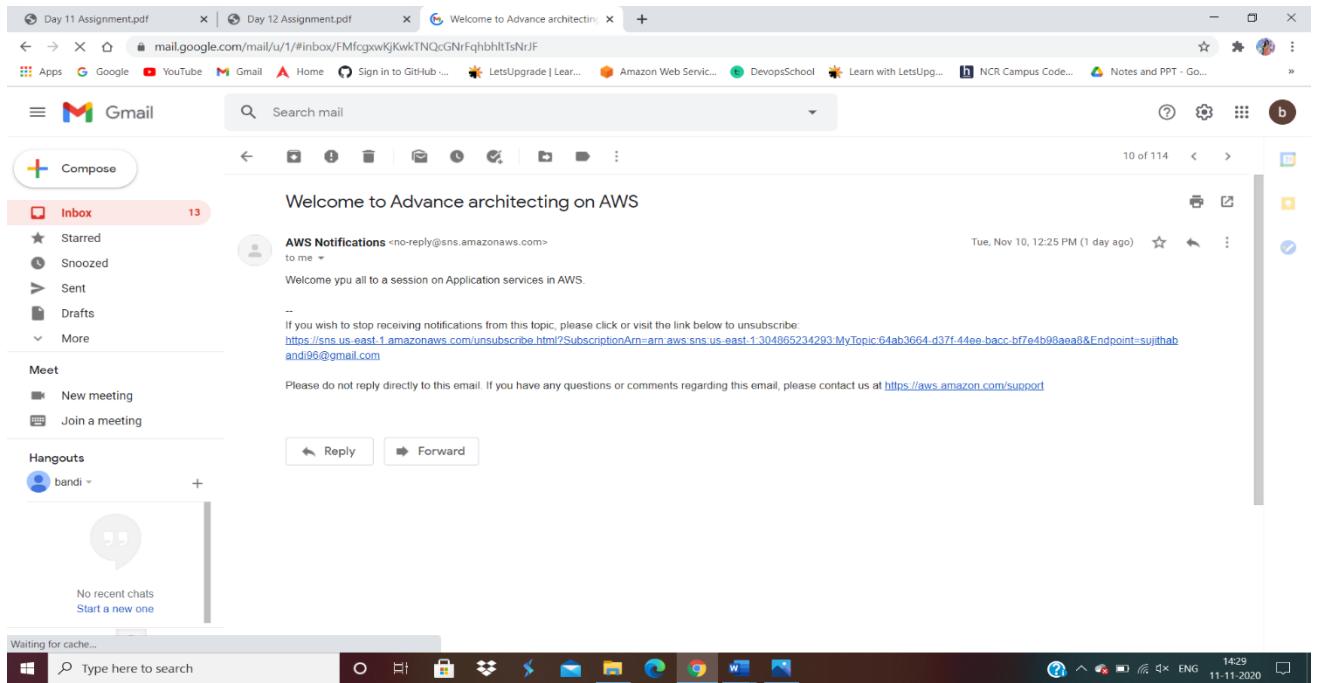
The screenshot shows the AWS SNS console with a green success message: "Topic MyTopic created successfully. You can create subscriptions and send messages to them from this topic." The "Topics" section is selected in the sidebar. The main details pane shows the topic name "MyTopic", ARN "arn:aws:sns:us-east-1:304865234293:MyTopic", and type "Standard". The "Subscriptions" tab is active, showing "Subscriptions (0)".

The screenshot shows a Gmail inbox with an email from "AWS Notifications" titled "AWS Notification - Subscription Confirmation". The email body states: "You have chosen to subscribe to the topic: arn:aws:sns:us-east-1:304865234293:MyTopic". It includes a link to "Confirm subscription". The Gmail interface shows other tabs like "Compose" and "Inbox (13)".

Ss2:Your inbox with the published message

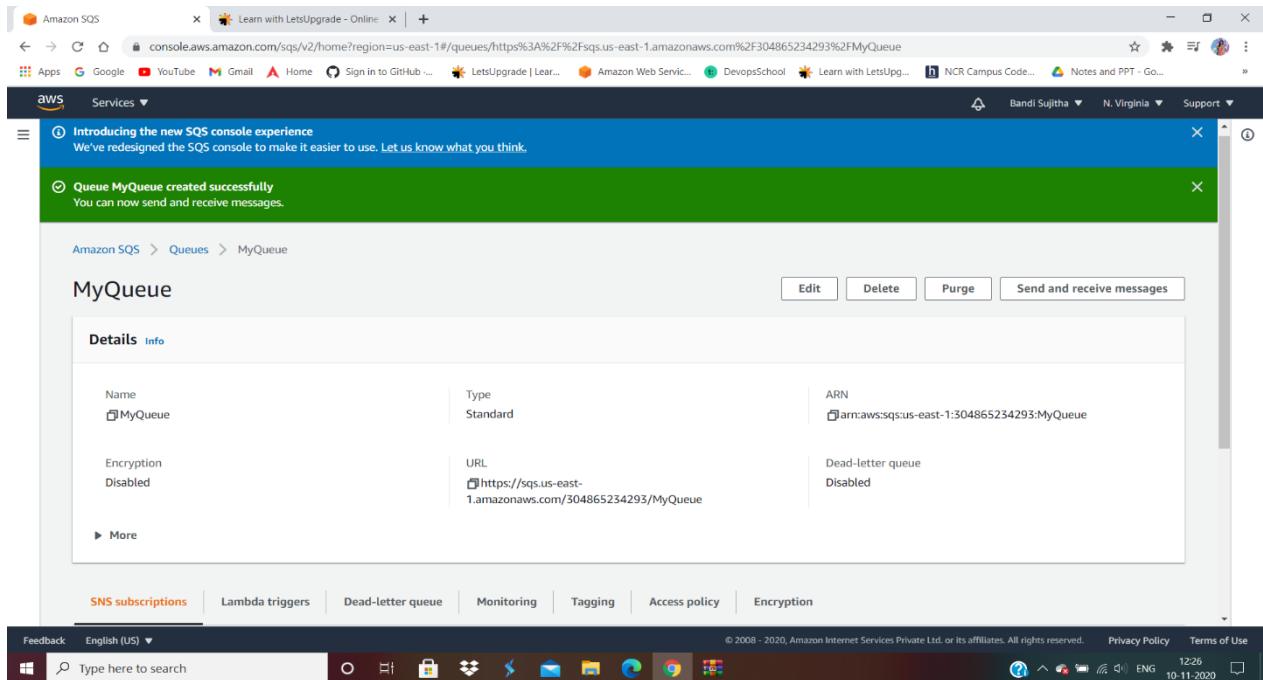
The screenshot shows the AWS SNS 'Publish message to topic' interface. At the top, the URL is `console.aws.amazon.com/sns/v3/home?region=us-east-1#/publish/topic/armaws sns:us-east-1:304865234293:MyTopic`. The page title is 'Publish message to topic'. The 'Message details' section includes a 'Topic ARN' field with the value `arn:aws:sns:us-east-1:304865234293:MyTopic`, a 'Subject - optional' field containing 'Welcome to Advance architecting on AWS', and a 'Time to Live (TTL) - optional' field set to 45 seconds. The 'Message body' section shows two delivery protocol options: 'Identical payload for all delivery protocols' (selected) and 'Custom payload for each delivery protocol'. The message body text area contains the text '1 Welcome you all to a session on Application services in AWS.' The bottom right corner shows the date and time as 10-11-2020 12:23 ENG.

This screenshot shows the same AWS SNS interface as above, but with additional message attributes visible. In the 'Message attributes' section, there are two entries: 'availability' with a value of 'yes' and 'Type' set to 'String', and another 'availability' entry with a value of '10' and 'Type' set to 'Number'. The rest of the interface is identical to the first screenshot, including the message body and delivery protocol selection.



Task 2 :Working with SQS

Ss1:SQS console



Ss2: Poll for messages and display message window

The screenshot shows the AWS SQS console with a message poller configuration. The 'Delivery delay' is set to 0 seconds. Under 'Receive messages', there are 2 messages available, with a polling duration of 30 seconds and a maximum message count of 10. The 'Polling progress' is at 2 receives/second. A table lists the received messages:

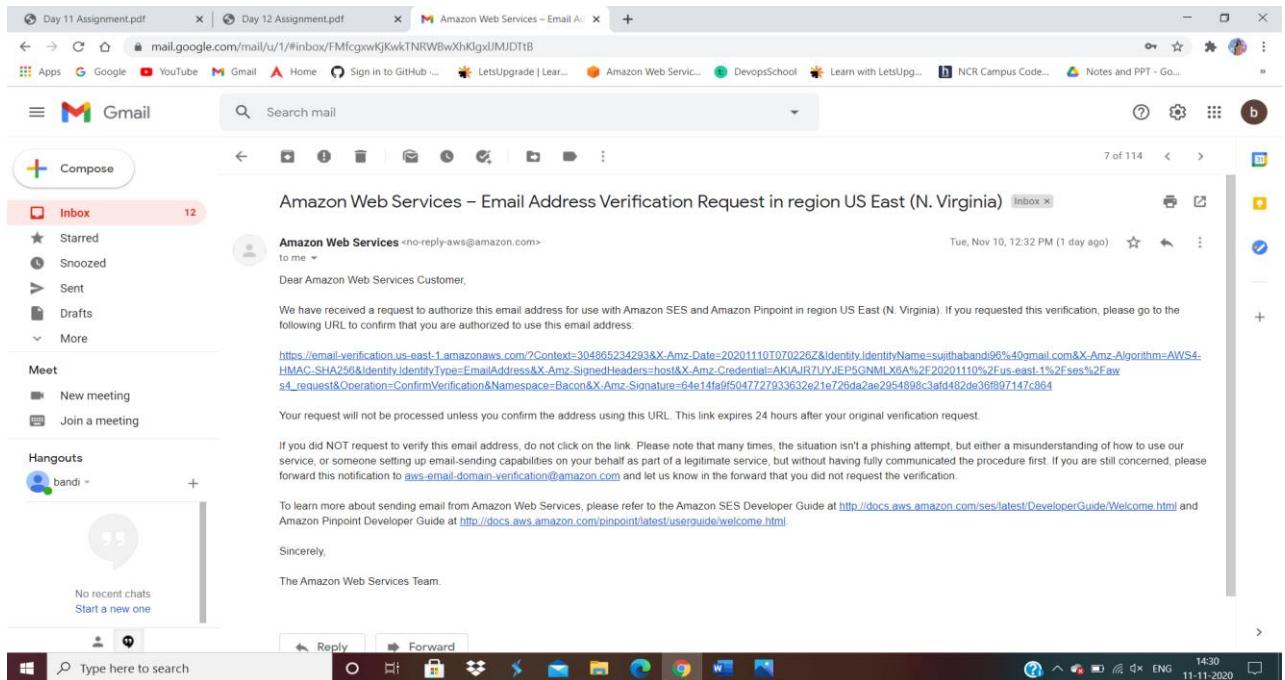
ID	Sent	Size	Receive count
b763f7f2-2e49-4fad-8a52-8f1de1b0fc22	11/10/2020, 12:30:33	960 bytes	1
cbf087f7-f210-475d-bba3-2f7f97c326fe	11/10/2020, 12:28:05	966 bytes	2

Task 3: Working with SES

Ss1: Inbox mail verification

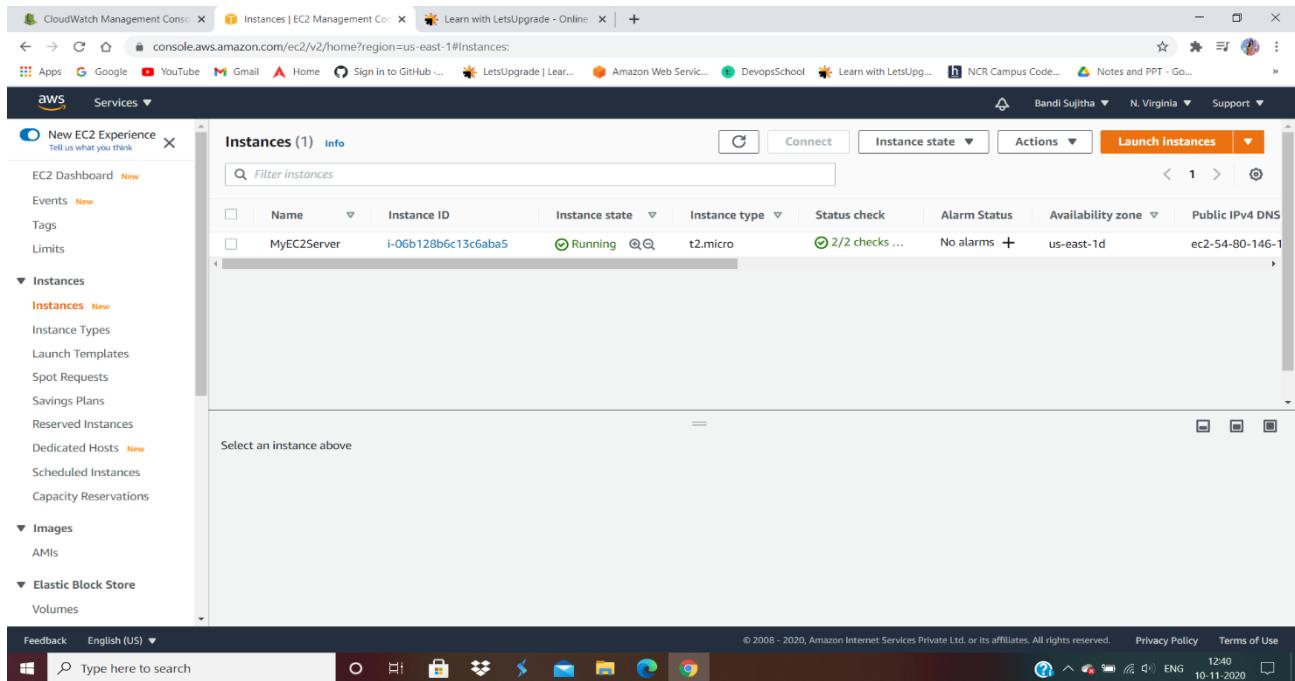
The screenshot shows the AWS SES Management Console under the 'Email Addresses' section. It displays a list of verified email identities:

Email Address Identities	Verification Status
saralabandi10@gmail.com	verified
sujithabandi96@gmail.com	verified



Task 4: Triggering CloudWatch event SNS notification

Ss1: EC2 console



Ss2: Rule configuration

The screenshot shows the 'Step 1: Create rule' page in the AWS CloudWatch Management Console. On the left, a sidebar lists various services like CloudWatch, Dashboards, Alarms, Metrics, and Rules. The 'Rules' section is currently selected. The main area is titled 'Step 1: Create rule' with the sub-instruction 'Create rules to invoke Targets based on Events happening in your AWS environment.' Below this, the 'Event Source' section is active, showing 'Event Pattern' selected. It displays a configuration for 'EC2' with 'Event Type' set to 'EC2 Instance State-change Notification'. Under 'Any state', 'Any instance' is selected. An 'Event Pattern Preview' shows the JSON: { "source": ["aws.ec2"] }. To the right, the 'Targets' section is shown, with 'SNS topic' selected and 'Topic*' set to 'MyServerMonitor'. There are options for 'Configure Input' including 'Matched event', 'Part of the matched event', 'Constant (JSON text)', and 'Input Transformer'. A 'Add target*' button is available.

The screenshot shows the 'Step 2: Configure rule details' page. The sidebar remains the same, with the 'Rules' section selected. The main area is titled 'Step 2: Configure rule details' under 'Rule definition'. It shows a 'Name*' field with 'MyEC2ChangeEvent' and a 'Description' field with 'MyEC2ChangeEvent'. A 'State' checkbox is checked and labeled 'Enabled'. A note below states: 'CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.' At the bottom, there are 'Cancel', 'Back', and 'Create rule' buttons. The status bar at the bottom indicates the date and time as '10-11-2020 12:40'.

Ss3: SNS topic

The screenshot shows the AWS SNS Management Console. A green success message at the top states: "Topic MyServerMonitor created successfully. You can create subscriptions and send messages to them from this topic." Below this, the "Topics" section lists "MyServerMonitor". The "Details" panel shows the following information:

Name	MyServerMonitor
ARN	arn:aws:sns:us-east-1:304865234293:MyServerMonitor
Type	Standard
Display name	MyServerMonitor
Topic owner	304865234293

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 selected, showing "Subscriptions (0)". There are buttons for "Edit", "Delete", "Request confirmation", "Confirm subscription", and "Create subscription".

The screenshot shows the Gmail inbox. Two emails are visible:

- AWS Notification - Subscription Confirmation** (Inbox) from no-reply@sns.amazonaws.com (to me) on Tue, Nov 10, 12:19 PM (1 day ago). The email body says: "You have chosen to subscribe to the topic: arn:aws:sns:us-east-1:304865234293:My Topic. To confirm this subscription, click or visit the link below (if this was in error no action is necessary): [Confirm subscription](#)". It also includes a note: "Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)".
- MyServerMonitor** (Inbox) from no-reply@sns.amazonaws.com (to me) on Tue, Nov 10, 12:36 PM (1 day ago). The email body says: "You have chosen to subscribe to the topic: arn:aws:sns:us-east-1:304865234293:MyServerMonitor".

Ss4: state change event notification in your mailbox

The screenshot shows the AWS CloudWatch Management Console with the EC2 service selected. In the top navigation bar, there are two green notifications: "Successfully stopped i-06b128b6c13c6aba5" and "Successfully started i-06b128b6c13c6aba5". Below this, the "Instances (1/1)" section displays a single instance named "MyEC2Server" with the ID "i-06b128b6c13c6aba5". The instance is currently in a "Pending" state, which is highlighted in red. Other details shown include the instance type "t2.micro", a "Status check" of "2/2 checks ...", and no alarms. The instance is located in the "us-east-1d" availability zone and has a public IPv4 DNS of "-".

The screenshot shows a Gmail inbox with an email from "MyServerMonitor <no-reply@sns.amazonaws.com>" titled "AVVS Notification Message". The email body contains an AWS CloudWatch Events notification message. It includes the following JSON payload:

```
{"version": "0", "id": "Sb0f9bd0-2386-acb-bf4e-84732770ee4b", "detail-type": "EC2 Instance State-change Notification", "source": "aws.ec2", "account": "304865234293", "time": "2020-11-10T07:10:55Z", "region": "us-east-1", "resources": ["arn:aws:ec2:us-east-1:304865234293:instance/i-06b128b6c13c6aba5"], "detail": {"instance-id": "i-06b128b6c13c6aba5", "state": "stopping"}}
```

Below the message, there is a link to unsubscribe: <https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:304865234293:MyServerMonitor:68a9c9ec-6dd0-4ebe-a5be-6794b0f3004&Endpoint=sjithabandla@gmail.com>.

The email also includes a note: "Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>".

Day-12 Assignment

Question 1:

Task 1: Create a dynamo db table with minimum two disaster recovery zones and verify replication.

Ss1: Disaster recovery regions with the table

The screenshot shows the AWS DynamoDB Global Tables console. On the left, there's a sidebar with options like Dashboard, Tables, Backups, Reserved capacity, Preferences, DAX, and Events. The main area is titled 'Albums' and shows a table of global replicas. The table has columns for Region Name, Status, Read capacity units, Write capacity units, Auto Scaling, and Endpoint. Three replicas are listed: Asia Pacific (Singapore) is active with 5 units each; Asia Pacific (Sydney) is active with 5 units each; and US East (Ohio) is active with 5 units each. The status bar at the bottom indicates it's from 10-11-2020 at 13:01.

Replication in Singapore:

The screenshot shows the AWS DynamoDB Overview console for the 'Albums' table in the ap-southeast-1 region. The left sidebar is identical to the previous screenshot. The main area shows the 'Overview' tab selected. It displays 'Stream details' which include Stream enabled (Yes), View type (New and old Images), and Latest stream ARN (am:aws:dynamodb:ap-southeast-1:304865234293:table/Albums/stream/2020-11-10T07:26:47.304). Below that is the 'Table details' section, which lists the Table name (Albums), Primary partition key (Actors (String)), Primary sort key (Songs (String)), Point-in-time recovery (DISABLED), and Encryption Type (DECRYPT). The status bar at the bottom indicates it's from 10-11-2020 at 12:57.

Replication in Sydney:

The screenshot shows the AWS DynamoDB console interface for the Sydney region. On the left, the navigation menu includes 'Dashboard', 'Tables' (which is selected), 'Backups', 'Reserved capacity', 'Preferences', 'DAX', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. A 'Create table' and 'Delete table' button are at the top. The main area displays the 'Albums' table details. A message banner at the top says: 'The new DynamoDB console is now available and has features such as data export to Amazon S3. In the redesigned DynamoDB console, you can export your table data from any point in time in the preceding 35 days to your data lake in Amazon S3 and perform analytics on the data. Try out the new console and tell us what you think.' Below this, the 'Overview' tab is selected, showing Stream details: 'Stream enabled: Yes', 'View type: New and old images', and 'Latest stream ARN: arn:aws:dynamodb:ap-southeast-2:304865234293:table/Albums/stream/2020-11-10T07:30:29.017'. The 'Table details' section shows the table name 'Albums', primary partition key 'Actors (String)', primary sort key 'Songs (String)', and point-in-time recovery status 'DISABLED Enable'. At the bottom, there's a search bar and standard browser controls.

Ss2:Home region with all items displayed

The screenshot shows the AWS DynamoDB console interface for the US East (Ohio) region. The navigation menu is identical to the previous screenshot. The main area displays the 'Albums' table details. A message banner at the top is identical. Below this, the 'Items' tab is selected, showing a list of items under 'Scan: [Table] Albums: Actors, Songs'. The items are listed in two columns: 'Actors' and 'Songs'. The data is as follows:

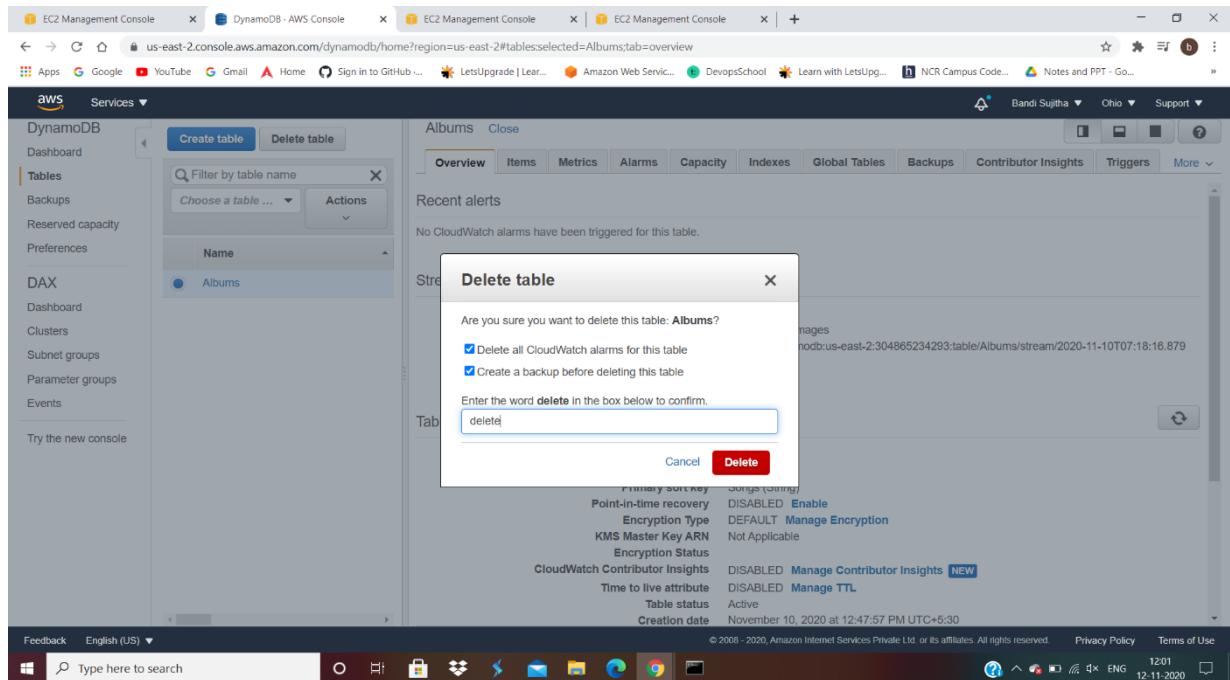
Actors	Songs
chiru	mava mava
rajini	maya nadhi
rajini	neruppu da
ram charan	jigelu rani
ram charan	ek baar ek baar

Ss3:Use query to fetch few items

The screenshot shows the AWS DynamoDB console interface. On the left, the navigation menu includes 'Dashboard', 'Tables' (selected), 'Backups', 'Reserved capacity', 'Preferences', 'DAX', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. A 'Create table' button is also present. The main area displays the 'Albums' table details. The 'Items' tab is selected, showing a query configuration for 'Query: [Table] Albums: Actors, Songs'. The 'key' field is set to 'Songs' with a sort key 'String'. The 'Sort' key is 'Songs' with 'Ascending' selected. The 'Attributes' section has 'Projected' selected. The results table shows two items: 'ram charan' under 'Actors' and 'ek baar ek baar' under 'Songs'. The status bar at the bottom indicates 'Viewing 1 to 2 items'.

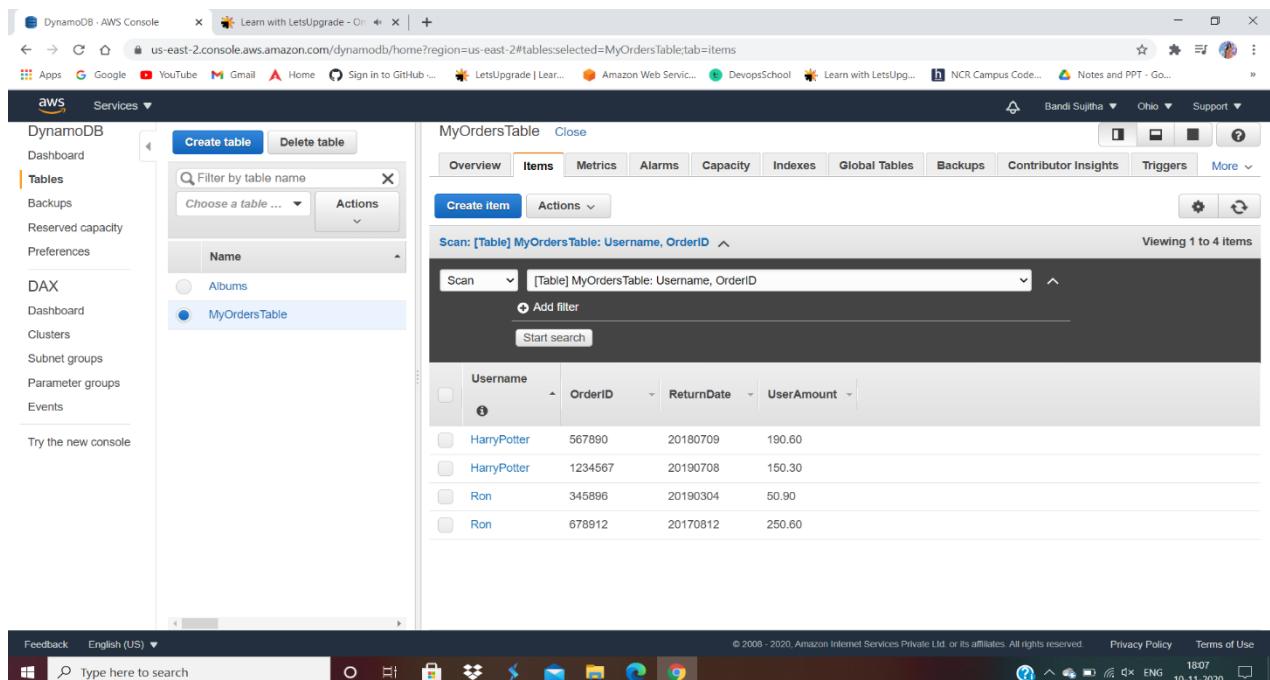
This screenshot is similar to the one above, but the 'Songs' value in the query is now set to 'ek bear ek baar'. The results table shows one item: 'ram charan' under 'Actors' and 'ek bear ek baar' under 'Songs'. The status bar at the bottom indicates 'Viewing 1 to 1 items'.

Ss4:Deletion and verification

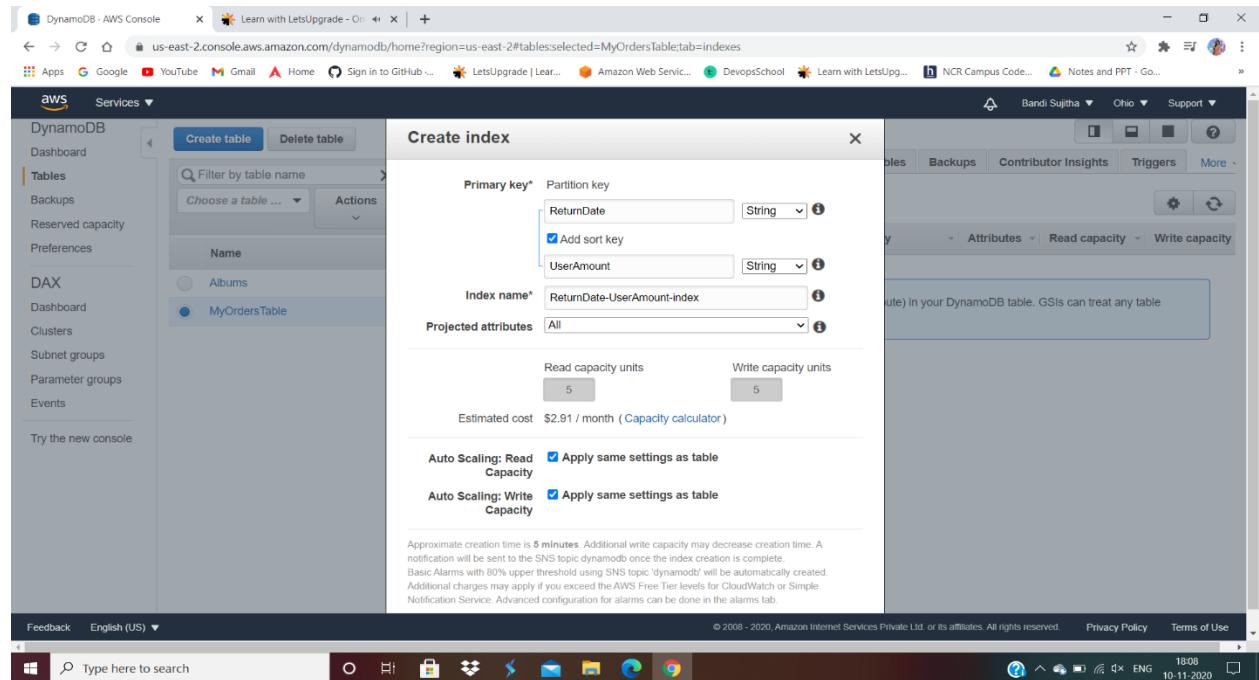


Task 2:Creating a dynamo DB table with global secondary indexes and fetching data using global secondary indexes.

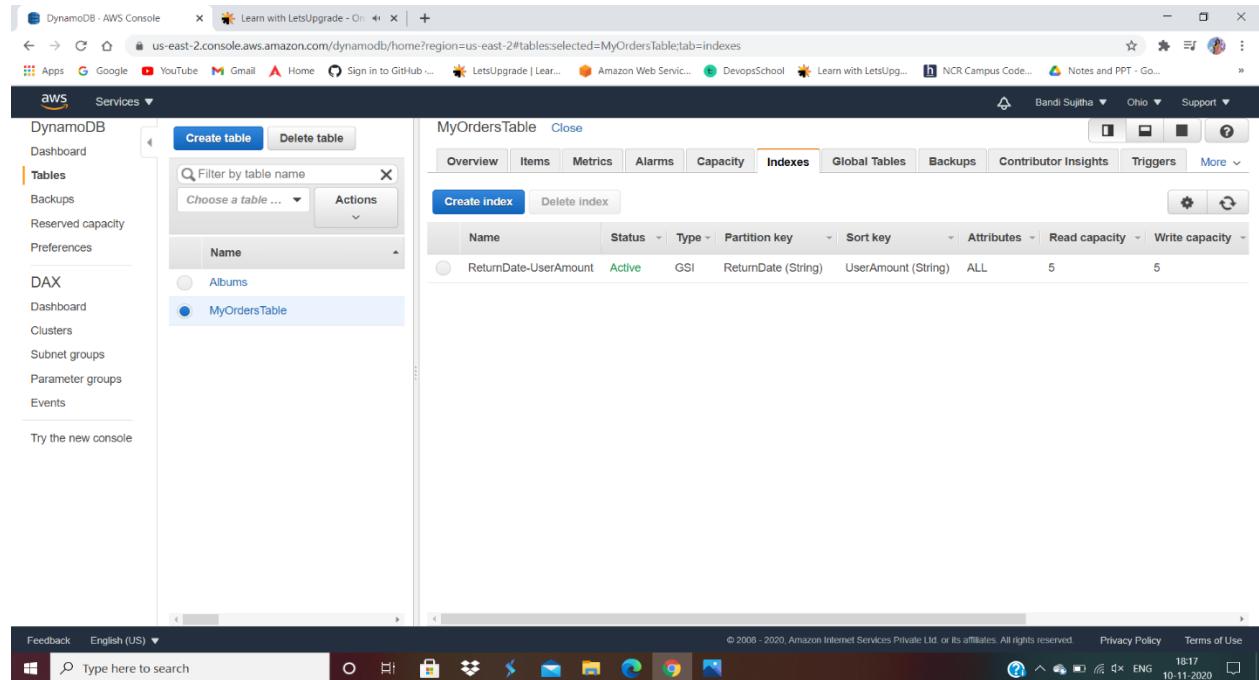
Ss1:Table with its items displayed



Ss2:Creating global secondary index



Ss3:Scan with global secondary index



DynamoDB - AWS Console

MyOrdersTable

Username	OrderID	ReturnDate	UserAmount
HarryPotter	567890	20180709	190.6
Ron	345896	20190304	50.9
Ron	678912	20170812	250.6

Task 3: Deploying a python application in elastic beanstalk

Ss1: Application page

Elastic Beanstalk Environments

Elastic Beanstalk Applications

Application name	Environments	Date created	Last modified	ARN
letsupgrade-pythonapp-env	LetupgradePythonapp-env	2020-11-11 19:18:59 UTC+0530	2020-11-11 19:18:59 UTC+0530	arn:aws:elasticbeanstalk:us-east-2:304865234293:application/letsupgrade-pythonapp

Ss2:env list page

The screenshot shows the AWS Elastic Beanstalk environments list page. The left sidebar shows 'Recent environments' with 'LetsupgradePythonapp-env-1'. The main area displays a table titled 'All environments' with two rows:

Environment name	Health	Application name	Date created	Last modified	URL	Running versions	Platform
LetsupgradePythonapp-env	ok	letsupgrade-pythonapp	2020-11-11 19:19:06 UTC+0530	2020-11-11 19:21:54 UTC+0530	LetsupgradePythonapp-env.eba-3sfpw4v.us-east-2.elasticbeanstalk.com	Sample Application	Python 3.7 running on 64bit Amazon Linux 2
LetsupgradePythonapp-env-1	ok	letsupgrade-pythonapp	2020-11-12 10:54:31 UTC+0530	2020-11-12 10:57:20 UTC+0530	reddomain123.us-east-2.elasticbeanstalk.com	letsupgrade-pythonapp-source-2	Python 3.7 running on 64bit Amazon Linux 2

Ss3:env health status page

The screenshot shows the AWS Elastic Beanstalk environment health status page for 'LetsupgradePythonapp-env-1'. The left sidebar shows 'letsupgrade-pythonapp' and 'LetsupgradePythonapp-env-1' with various options like 'Configuration', 'Logs', and 'Health'. A note at the top states: 'In September 2020, Elastic Beanstalk introduced the EnhancedHealthAuthEnabled option. It enables you to require authorization of instances that report enhanced health information. If you're using an Elastic Beanstalk managed policy for your environment's instance profile (the default when using Elastic Beanstalk console or EB CLI), you can safely enable this option.' Below this, another note says: 'On November 30, 2020, we plan on enabling this option by default for all new environments (no impact on existing environments). On May 31, 2021, we plan to start enforcing enhanced health authorization; it will be enabled for all new and existing environments, with no option to disable it.' A third note states: 'If you're using a custom instance profile, your environment might be impacted and might need a configuration update. To learn more, see Enhanced health authorization in the AWS Elastic Beanstalk Developer Guide.'

LetsupgradePythonapp-env-1
reddomain123.us-east-2.elasticbeanstalk.com (e-4akfkzkb2e)
Application name: letsupgrade-pythonapp

Health	Running version	Platform
Ok Causes	letsupgrade-pythonapp-source-2 Upload and deploy	Python 3.7 running on 64bit Amazon Linux 2/3.1.3 Change

Recent events