Software Engineer for Cloud Project 2

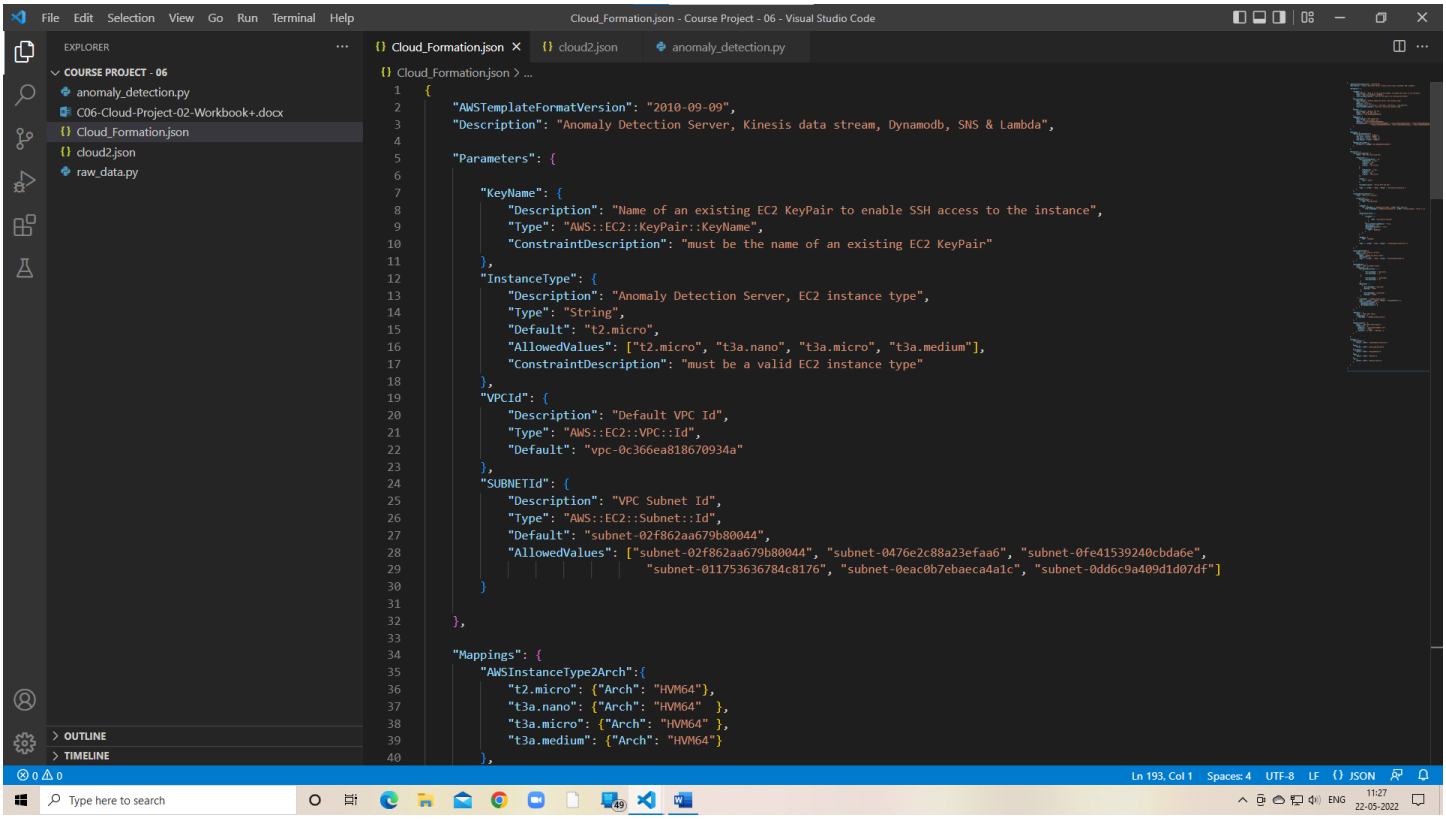
Anomaly Detection using CloudFormation and Codedeploy

Tasks with Screenshots

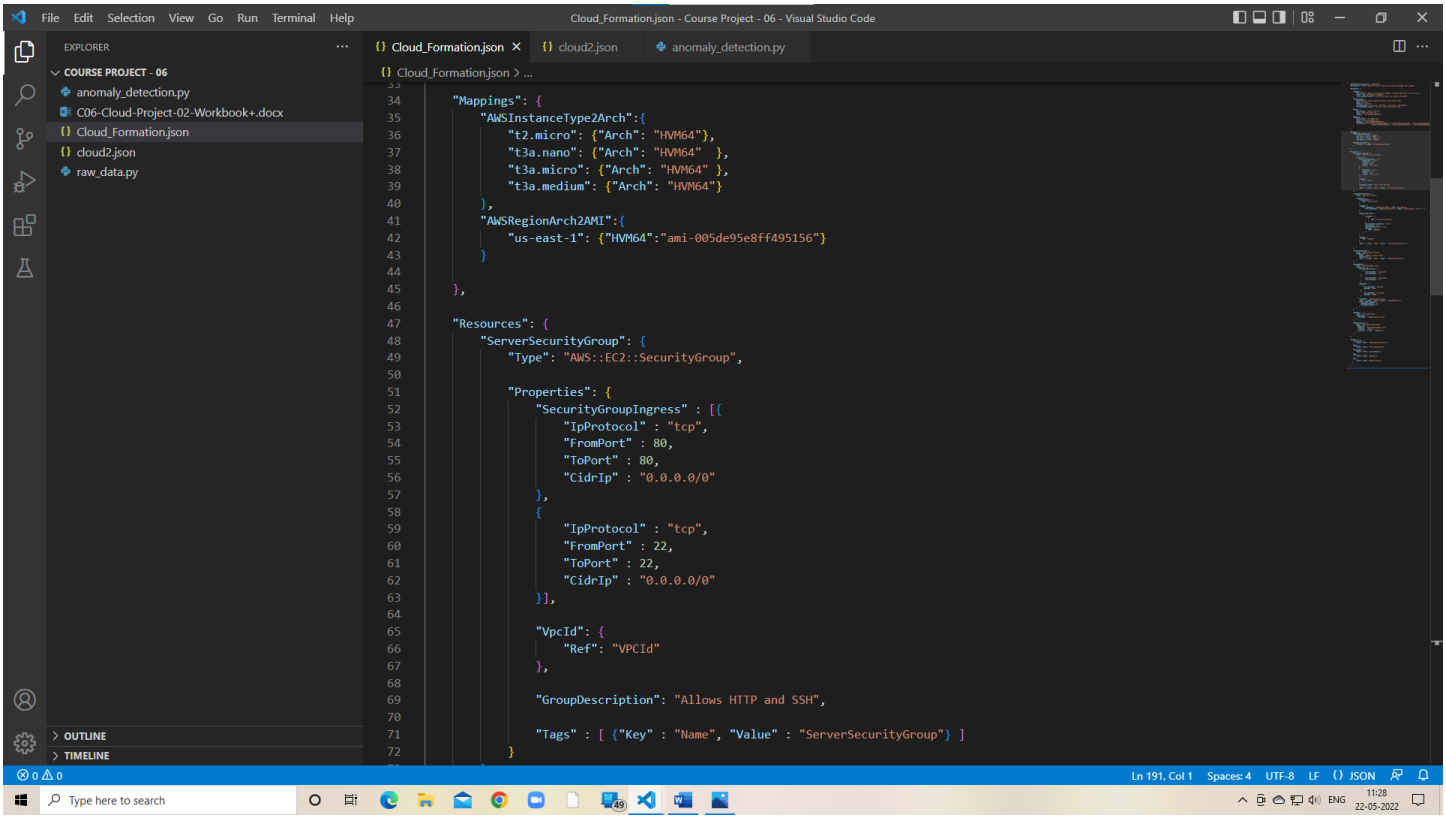
Task 1 Screenshots:

Step number	a
Step name	Cloud formation summary page
Instructions	1) Template file (JSON script) to enable mentioned services in the Task -1
Expected screenshots	1. Cloud formation summary page

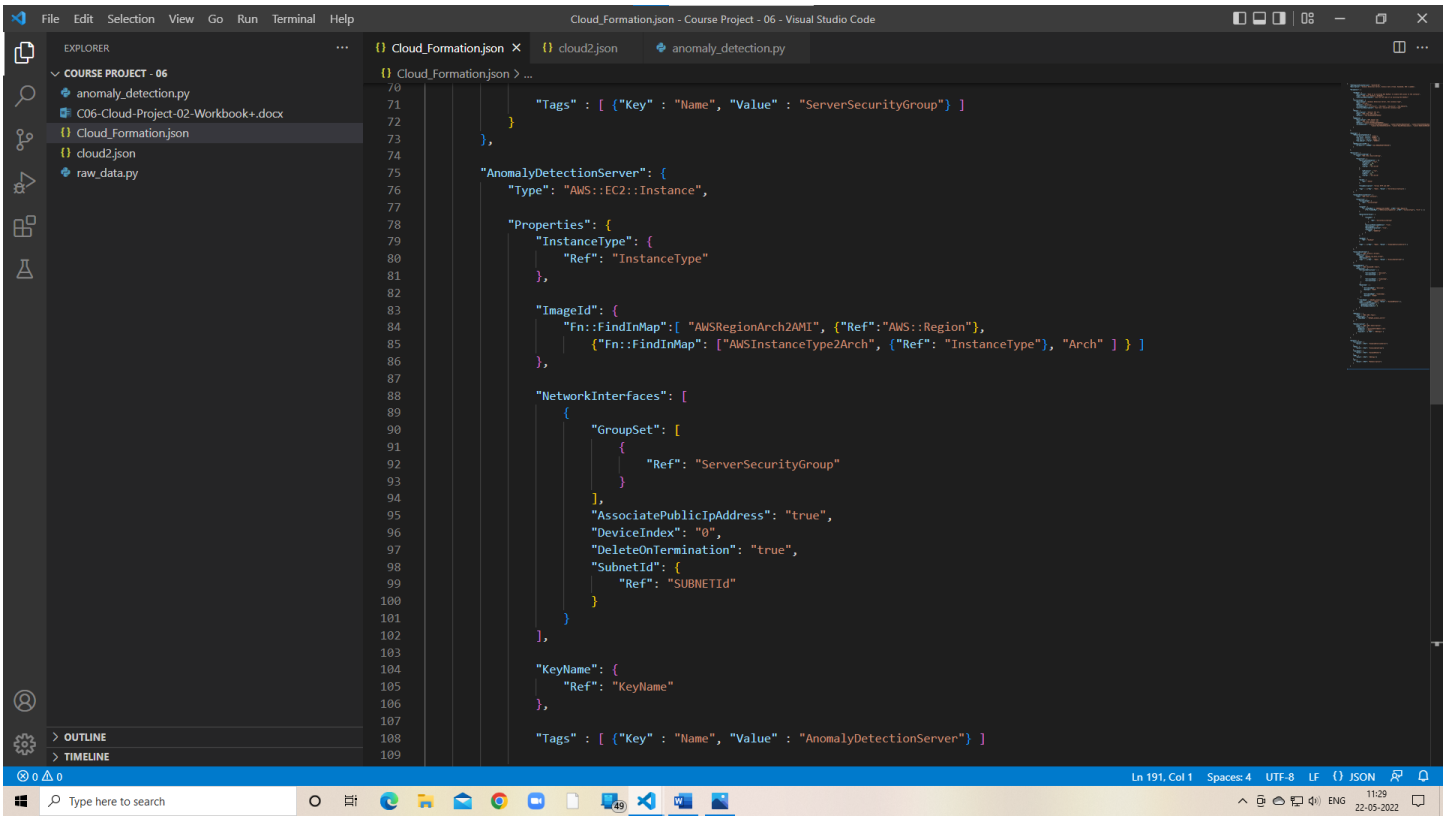
<Screenshot of Cloud_Formation.json template for TASK 1>

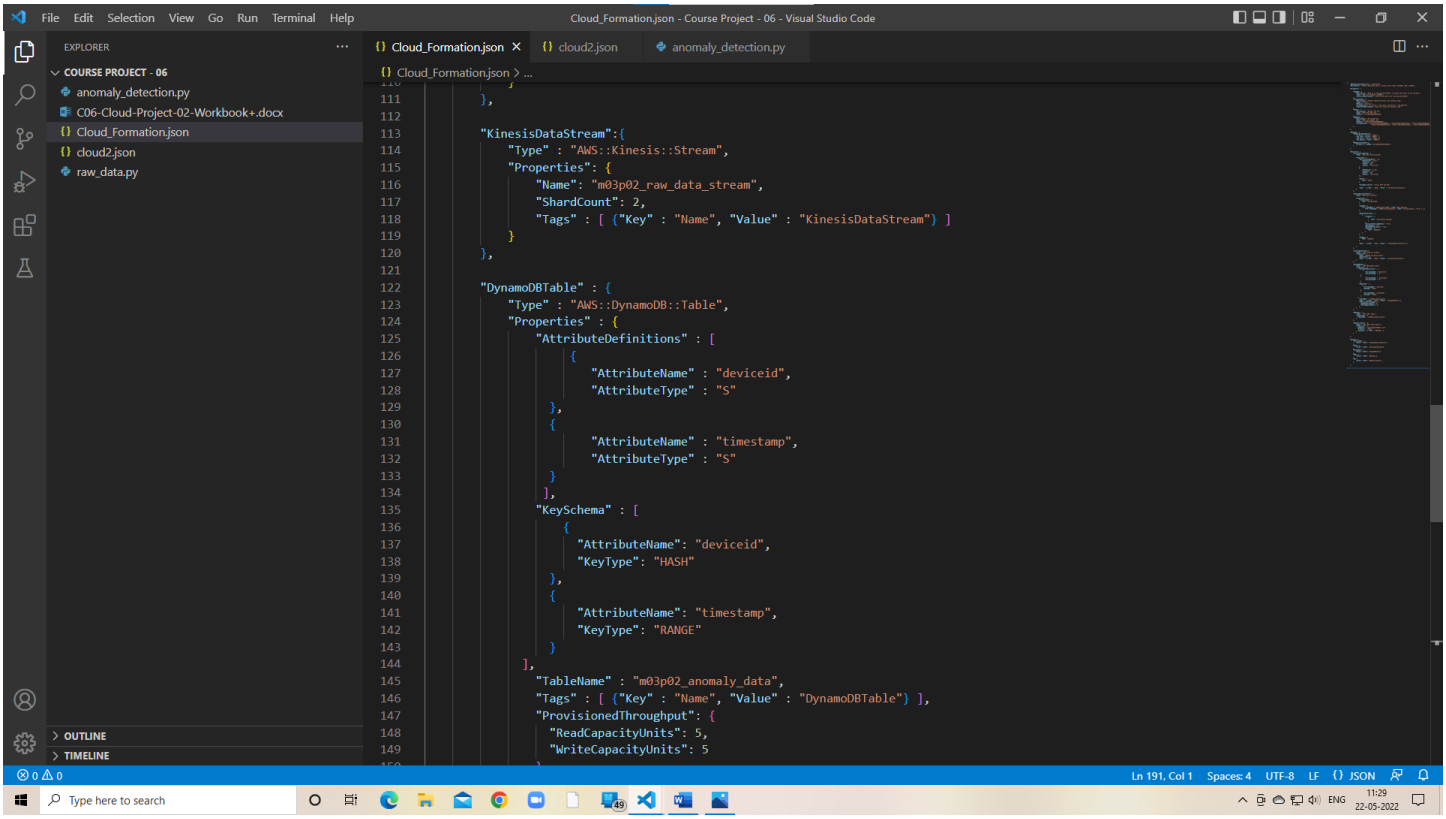


<Screenshot of Cloud_Formation.json template for TASK 1>

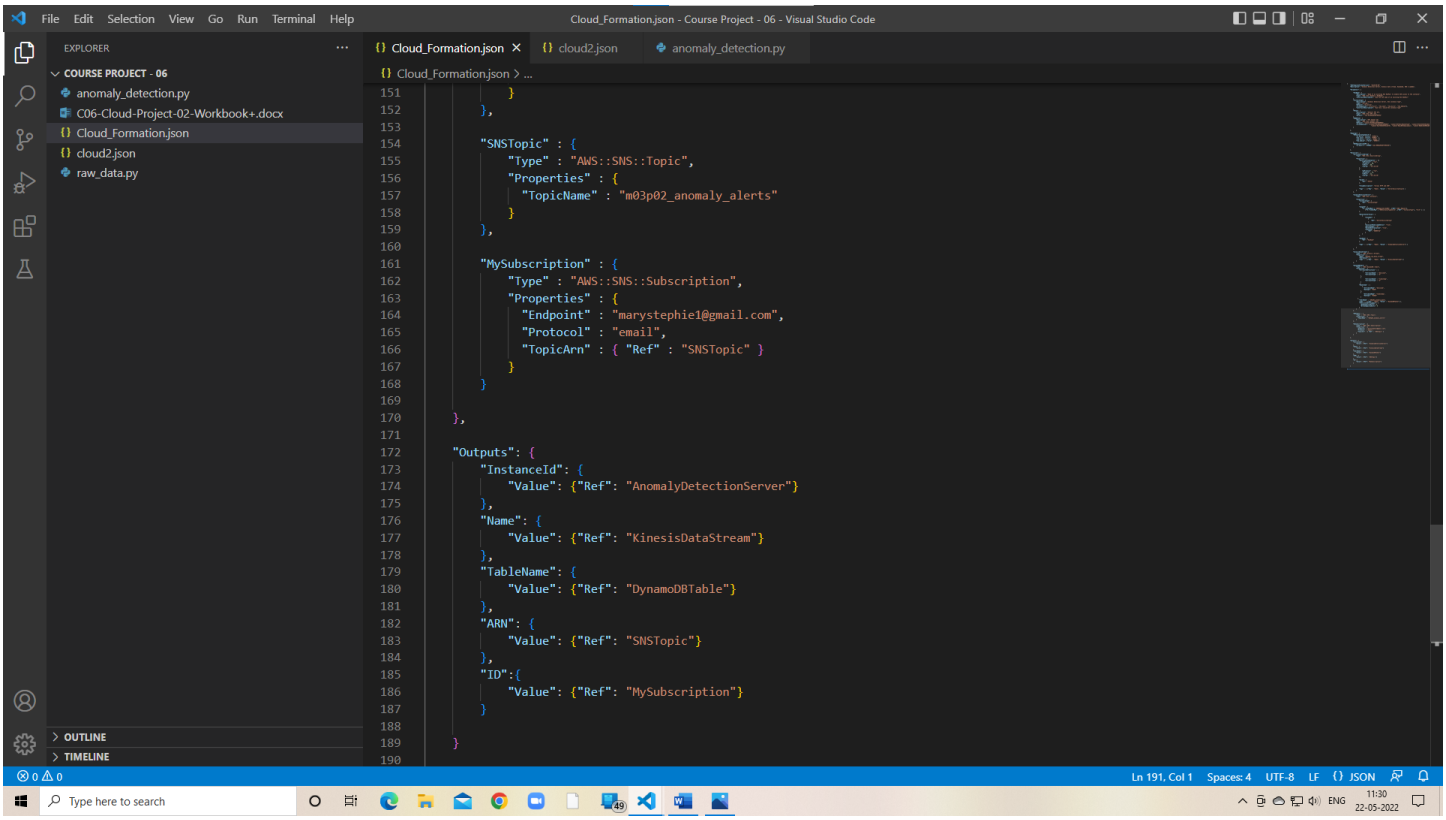


<Screenshot of Cloud_Formation.json template for TASK 1>

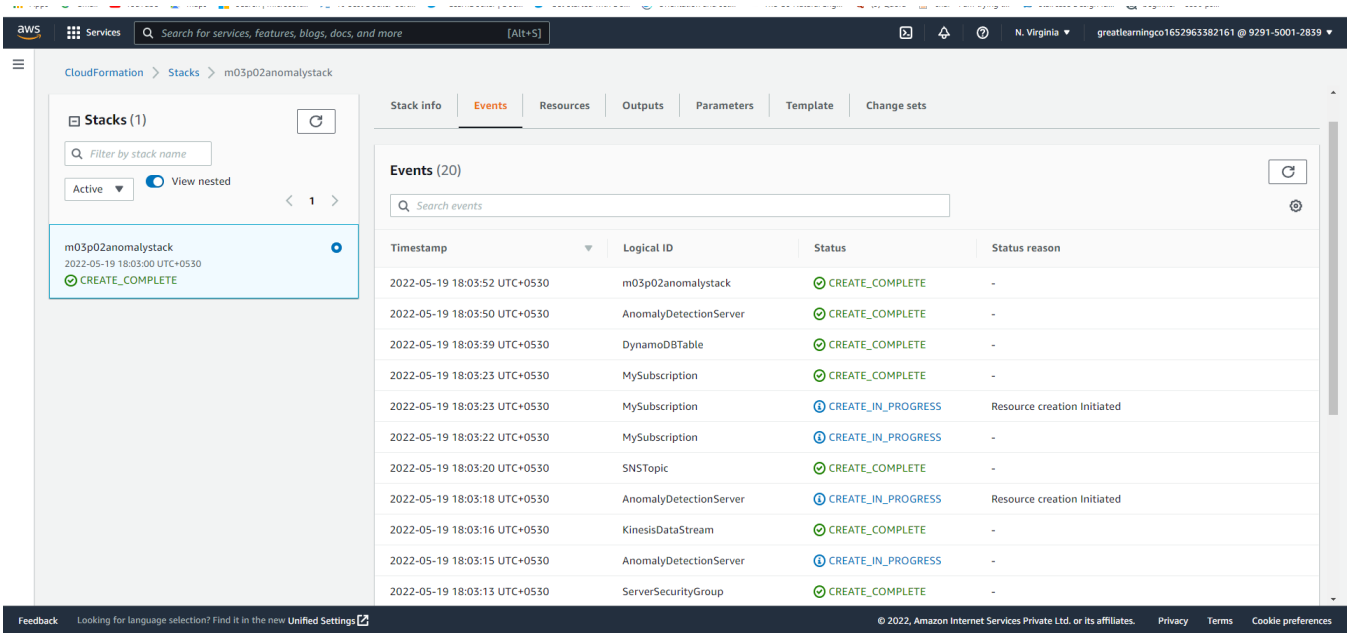




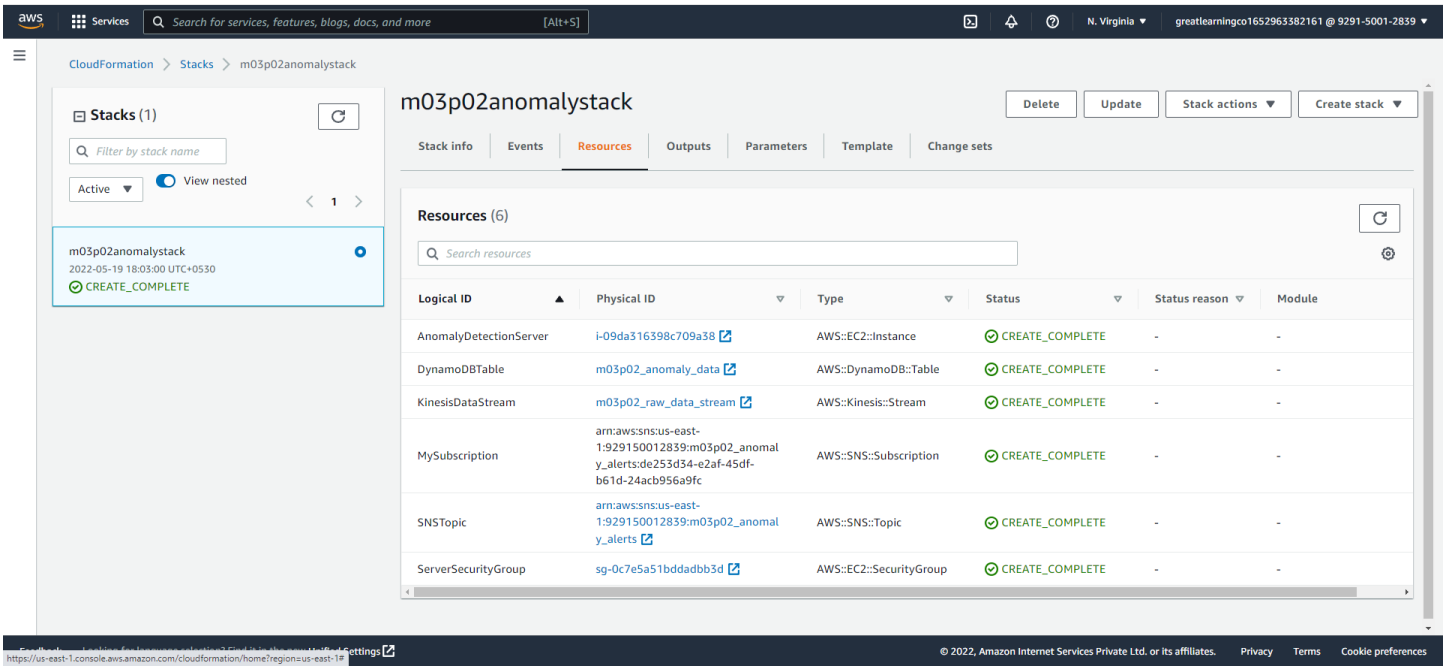
<Screenshot of Cloud_Formation.json template for TASK 1>



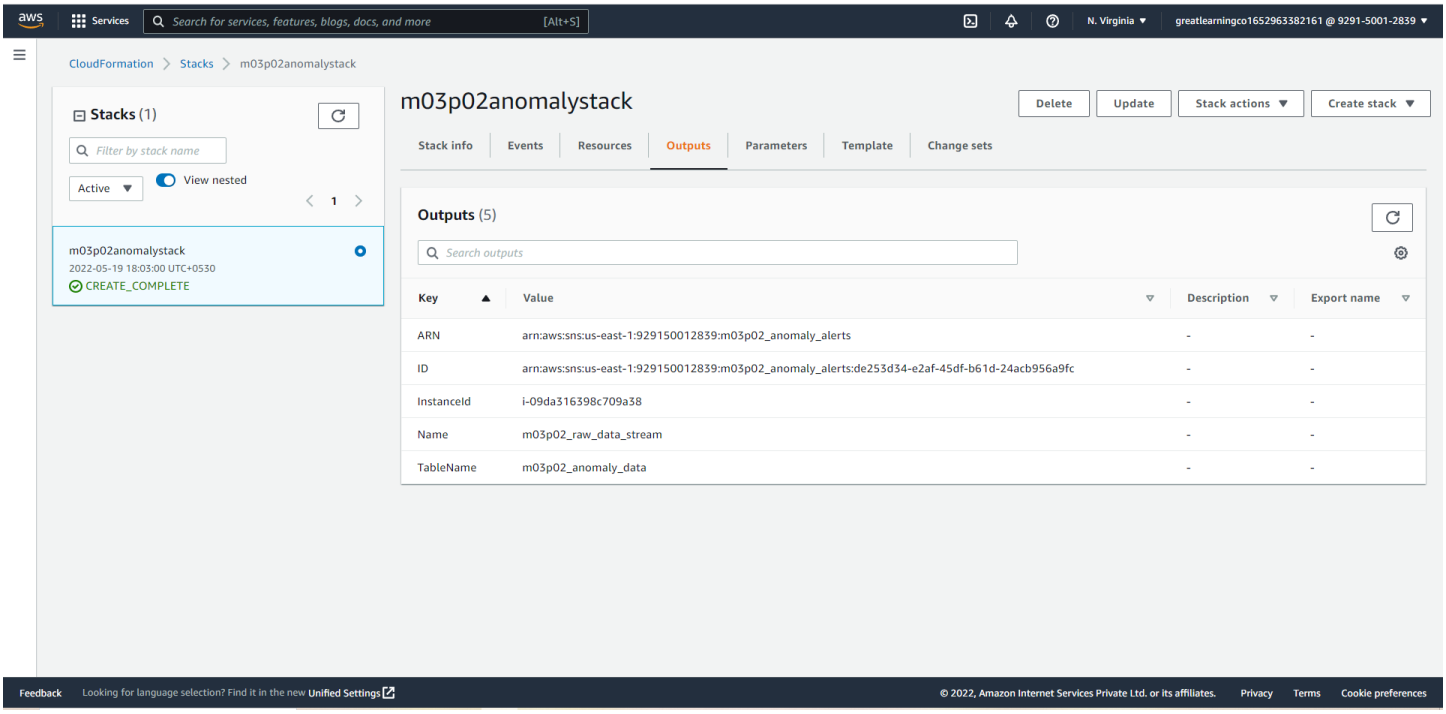
<Screenshot of a (1) Cloud Formation summary page >
<Creation of “m03p02stack” successful >



< “m03p02stack” Resources successful creation>

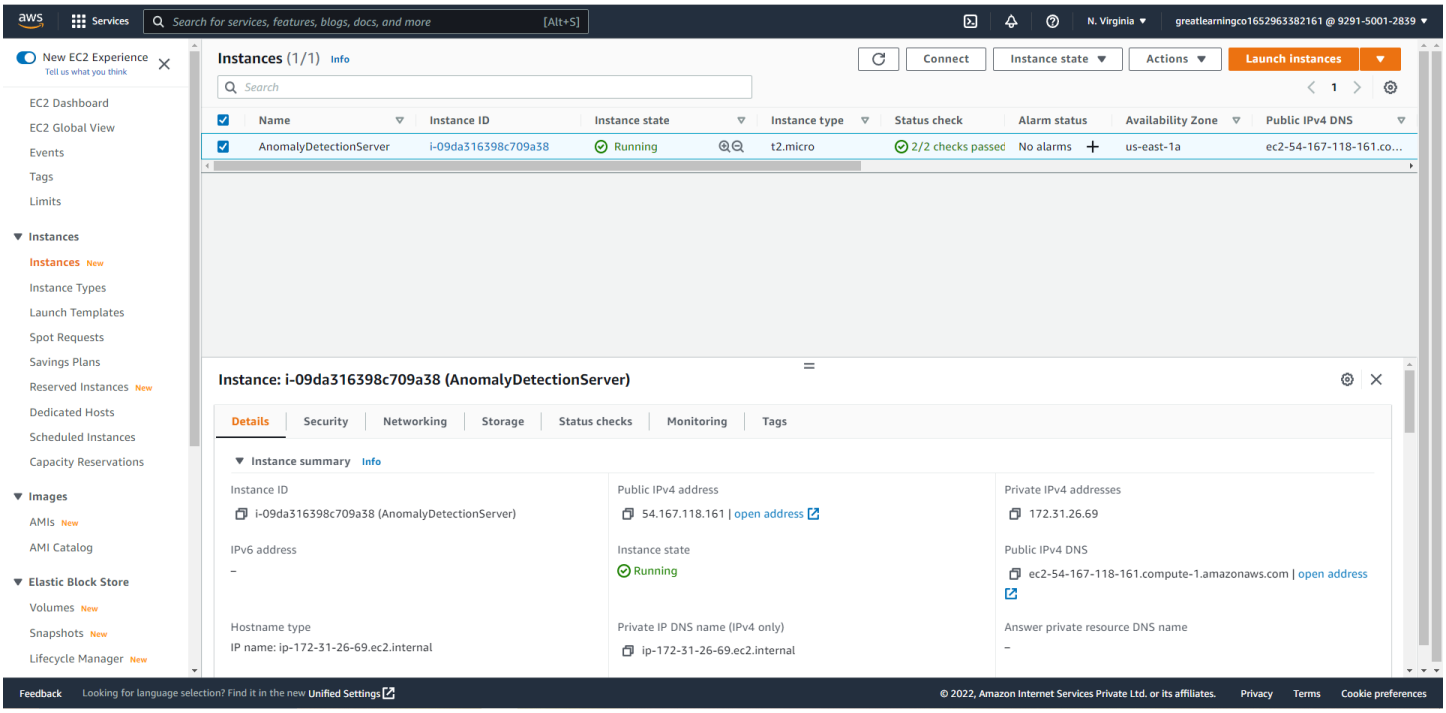


< “m03p02stack” Outputs summary page>

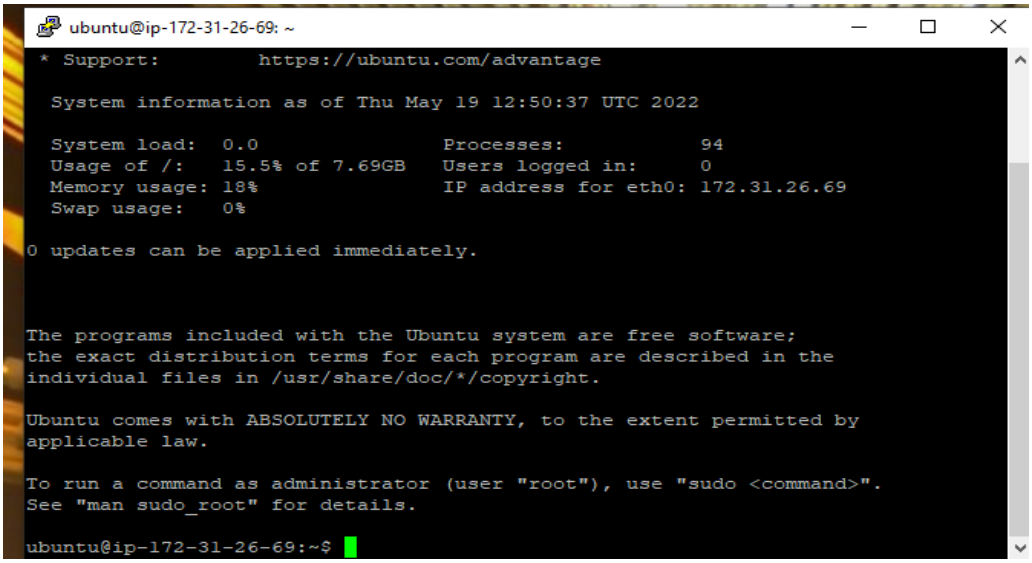


Step number	b
Step name	Enabled EC2 instance
Instructions	1) Make sure that cloud formation operation is successful 2) Check if the EC2 services are successful, if yes, check for the created instance on EC2 console
Expected screenshots	1) Created EC2 instance

< Screenshot b (1), EC2 Instance (Ubuntu server)>

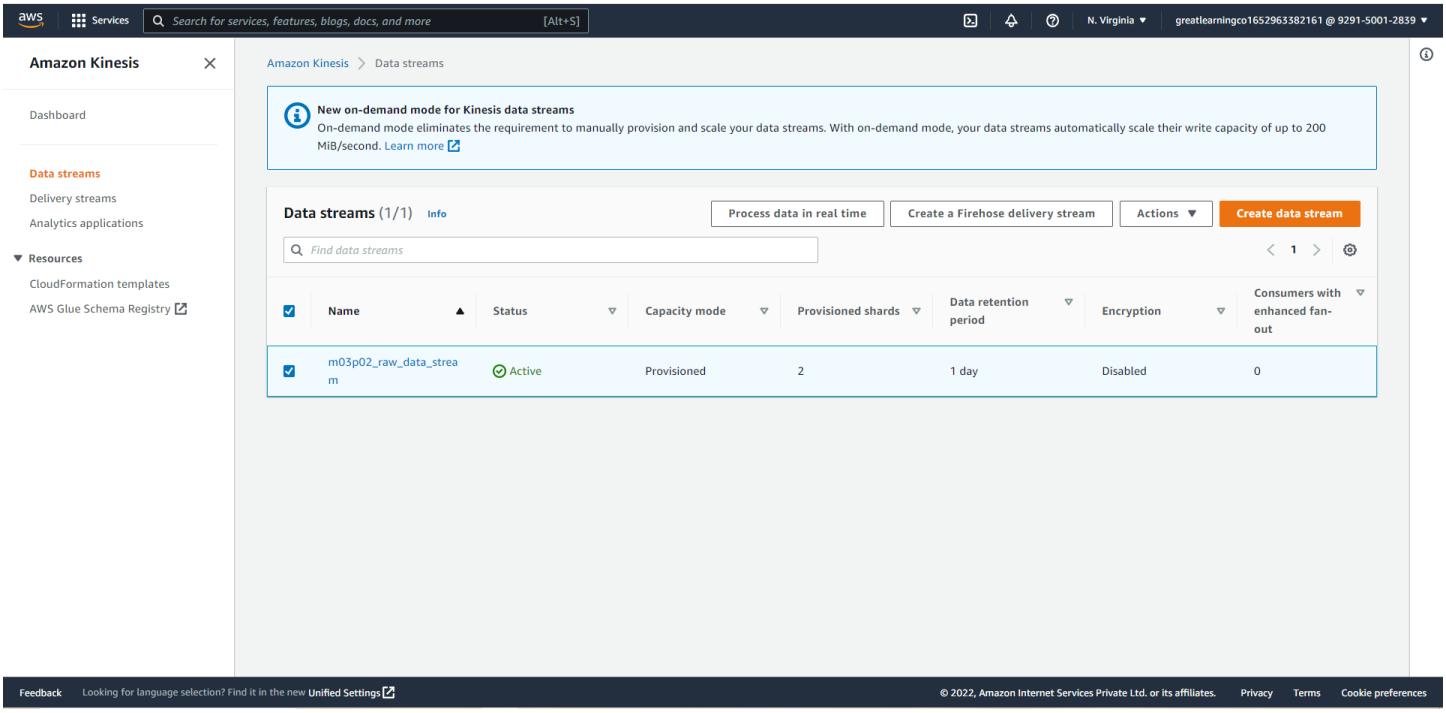


< EC2 Instance logged in via putty>

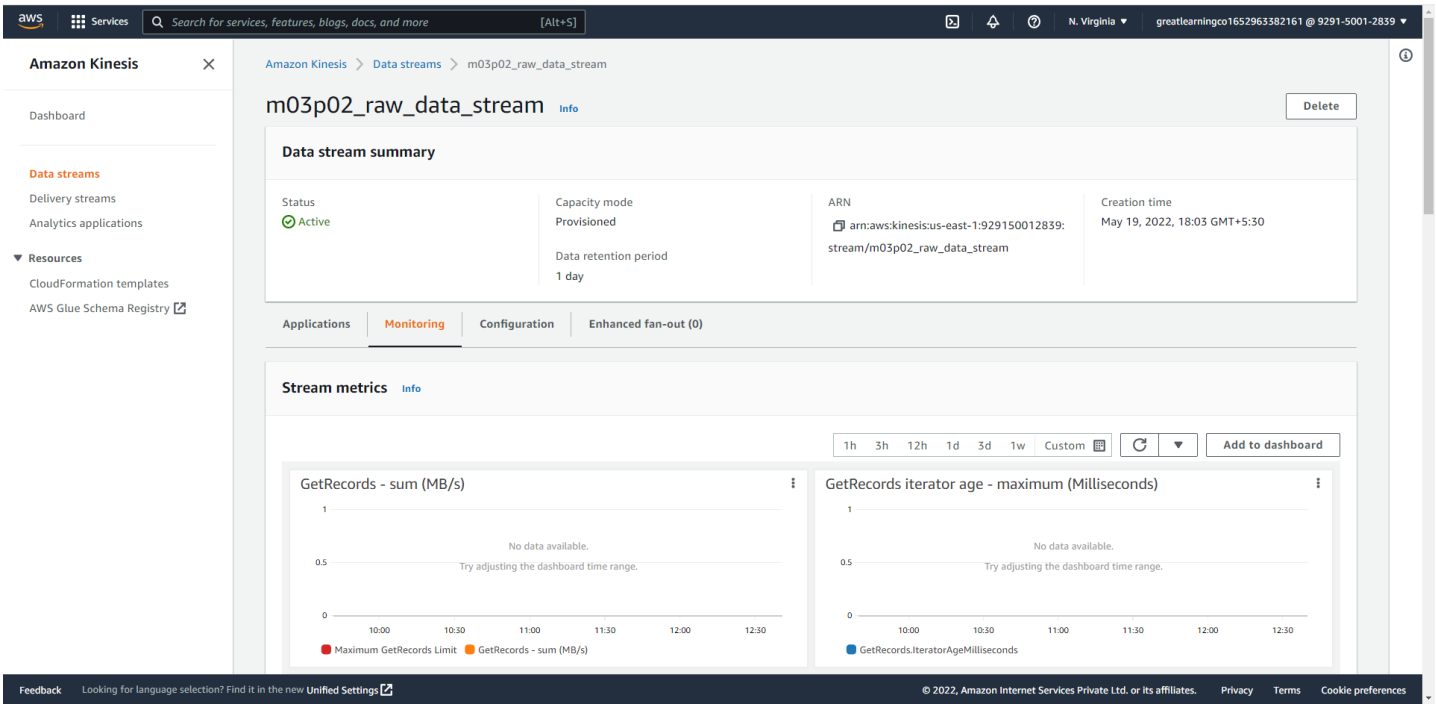


Step number	c
Step name	Enabled Kinesis stream
Instructions	1) Make sure that cloud formation operation is successful 2) Check if the kinesis services are successful, if yes, check for the created data stream on kinesis console
Expected screenshots	1) Create kinesis data stream

< c (1) Created Kinesis Data Stream>

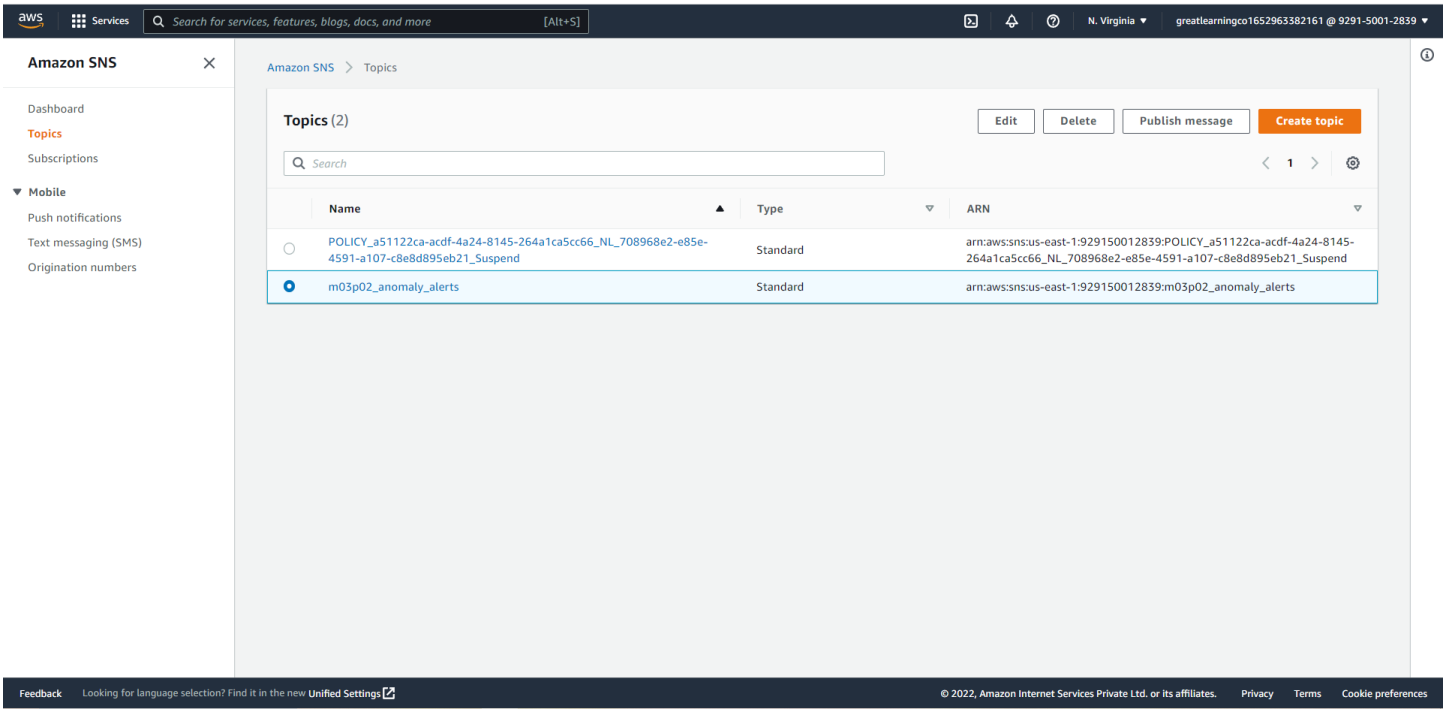


< c (1) Created Kinesis Data Stream>

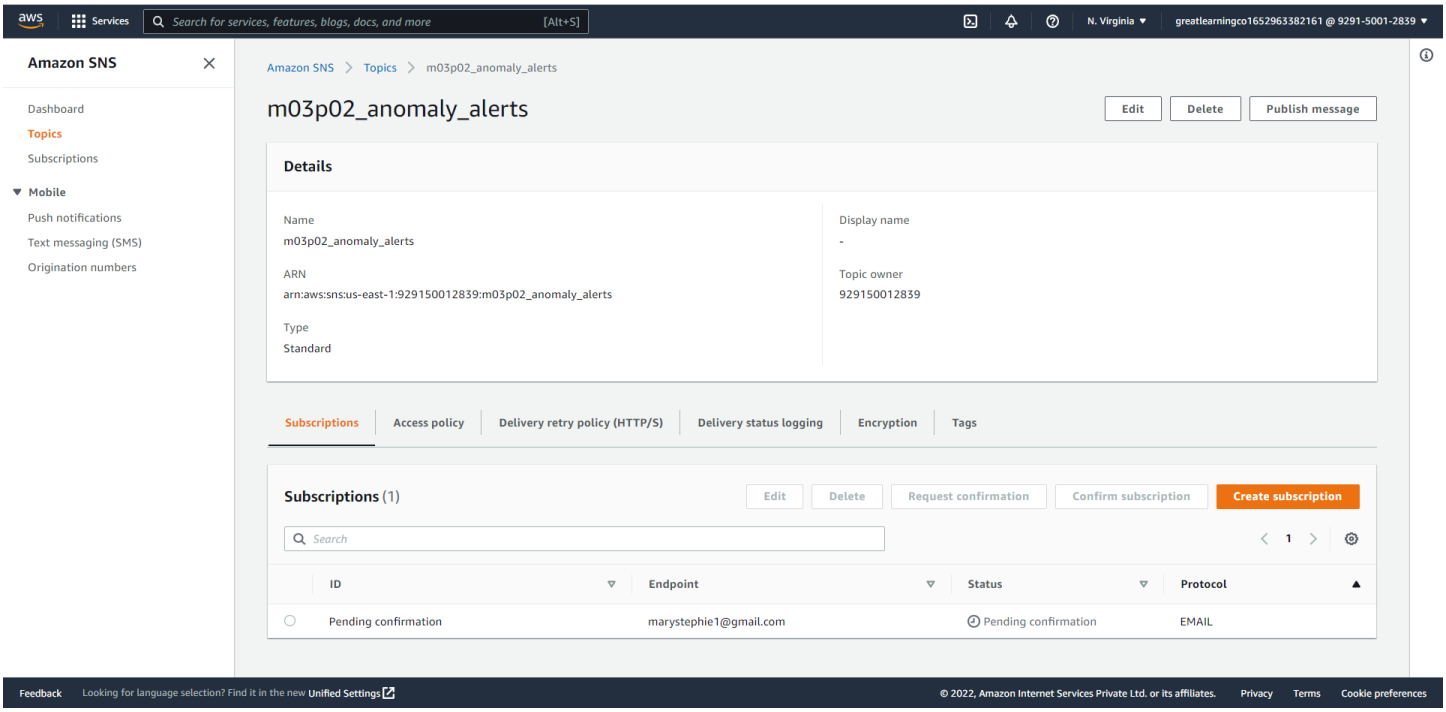


Step number	d
Step name	Enabled SNS arn
Instructions	1) Make sure that cloud formation operation is successful 2) Check if the SNS service are successful, if yes, goto SNS console and create a subscription for the created topic/arn
Expected screenshots	1. Created SNS arn 2. Created subscription and accepted subscription through mail

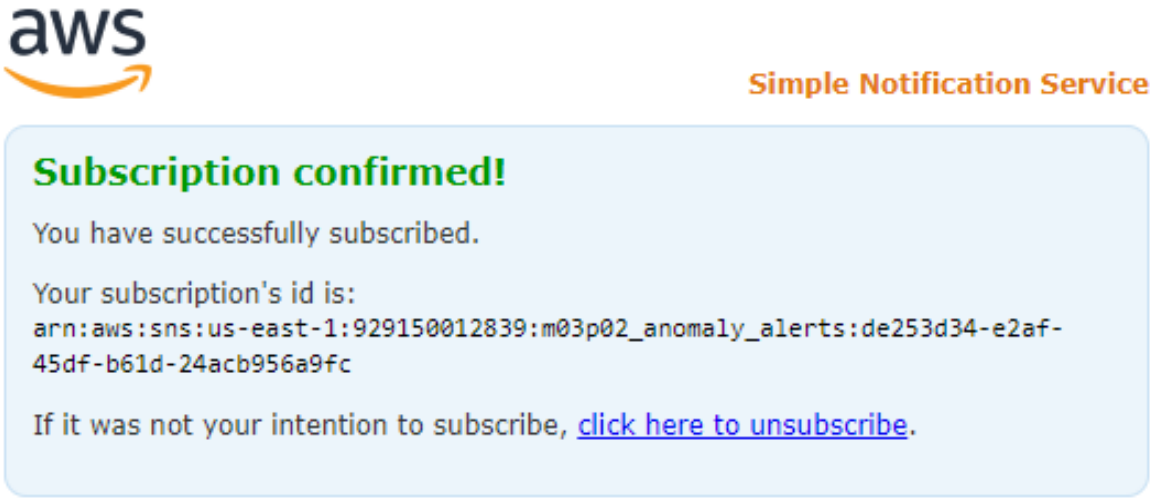
< Screenshot d (1) created SNS arn and SNS topic m03p02_anomaly_alerts >



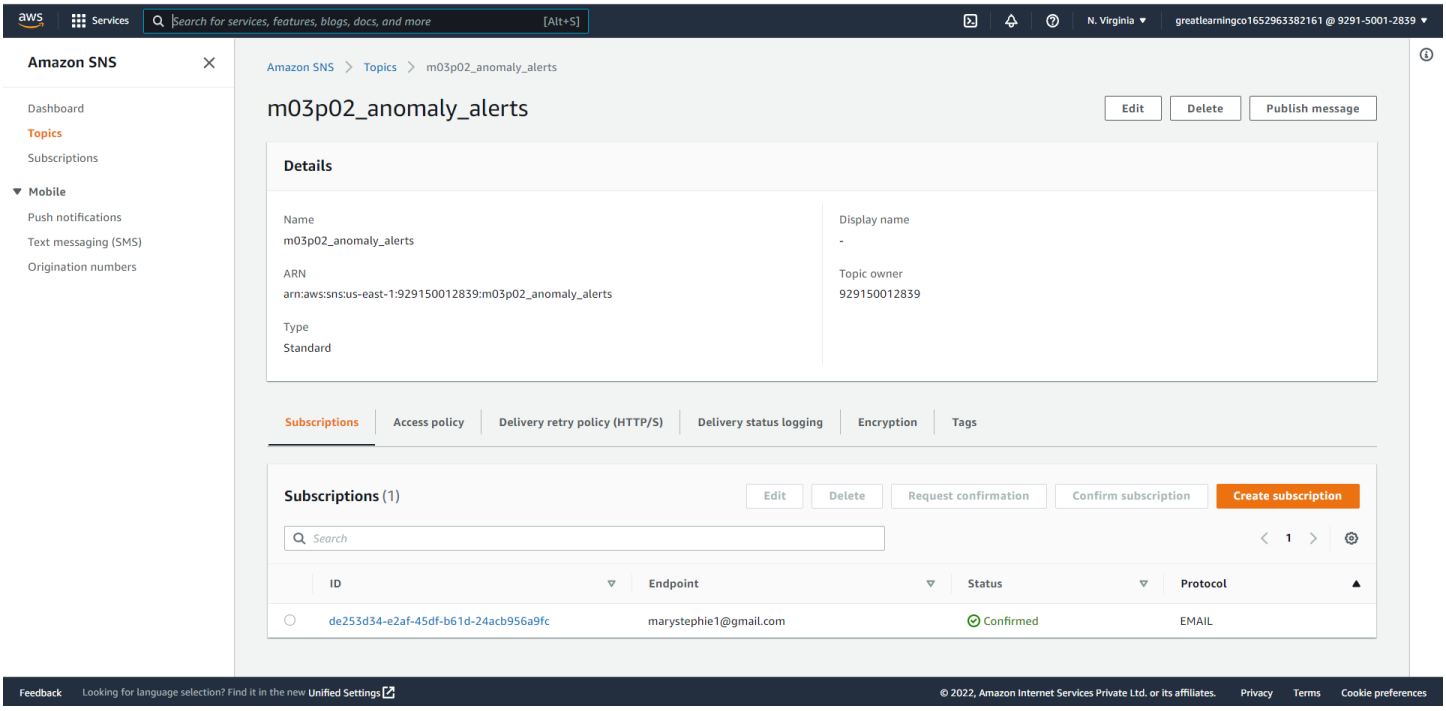
< SNS topic m03p02_anomaly_alerts and subscription >



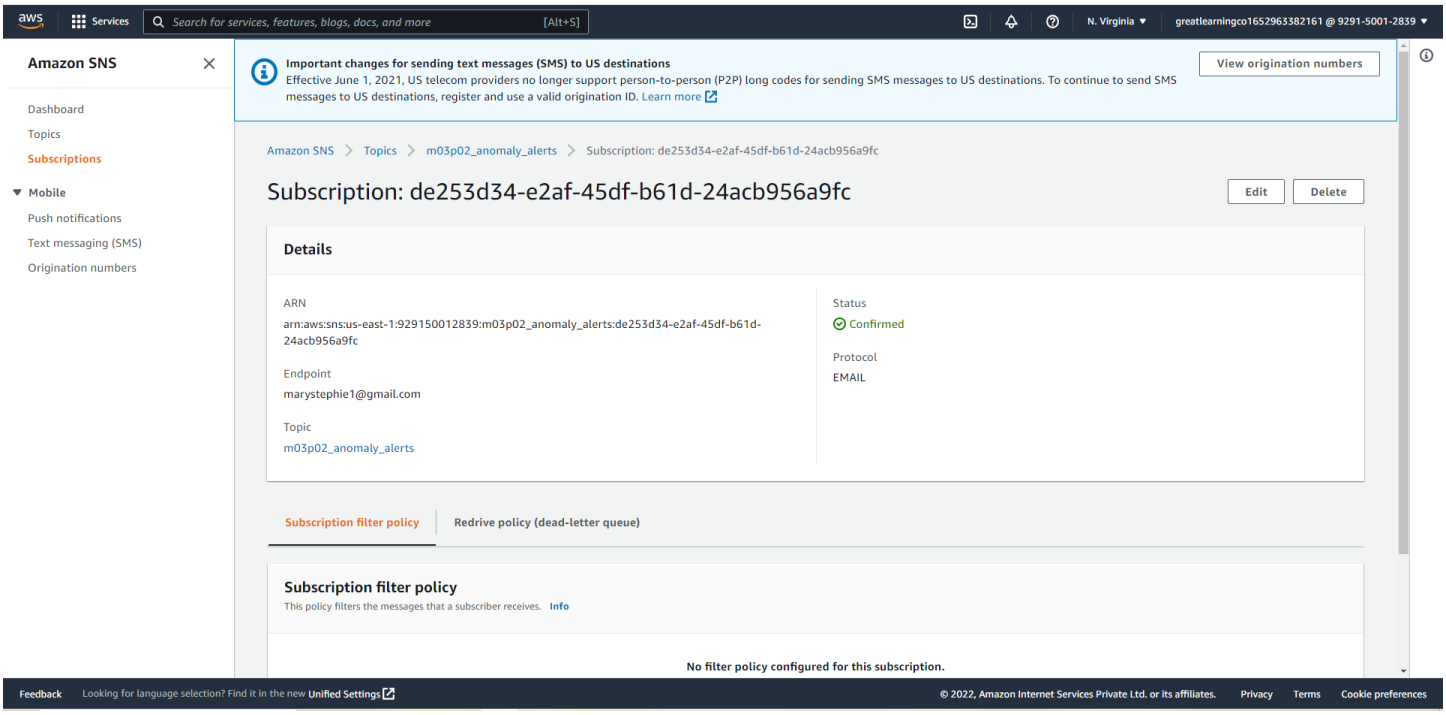
<Screenshot d (2) created subscription confirmed through mail >



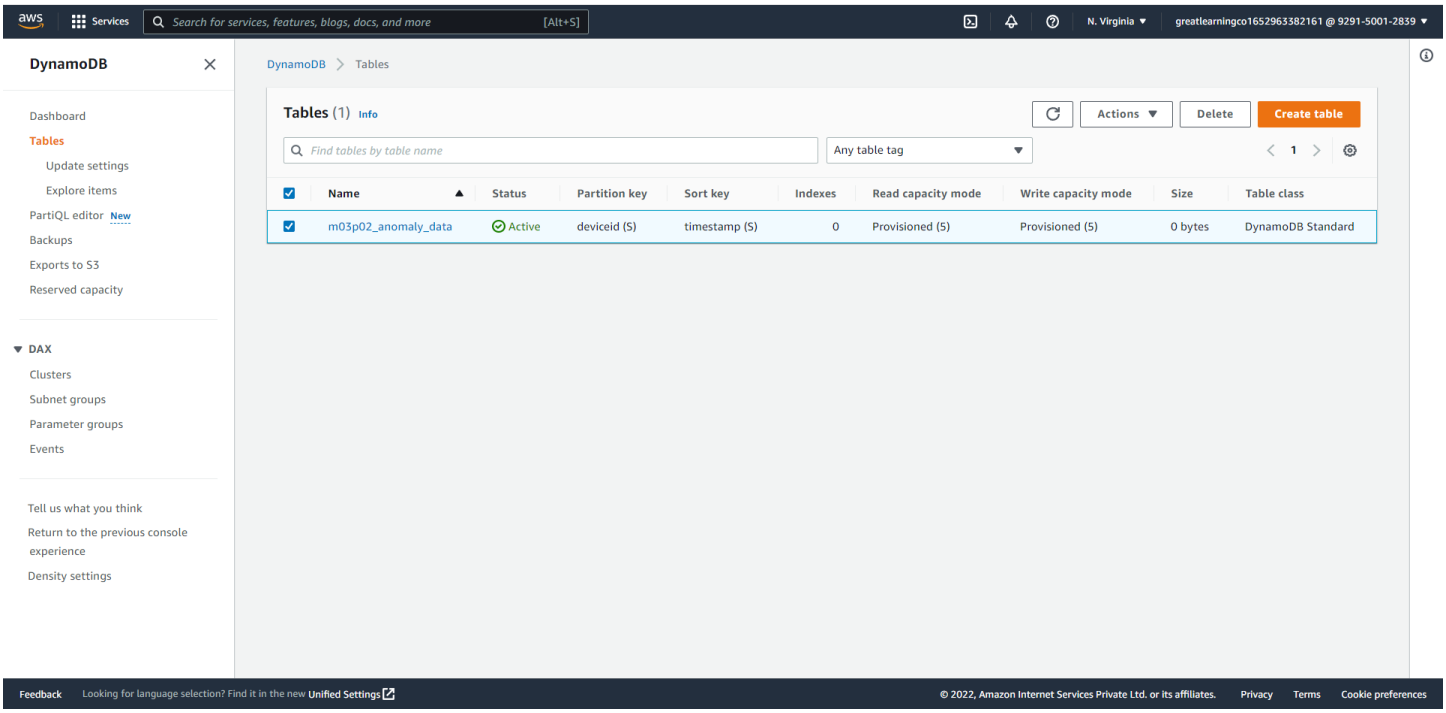
<Screenshot d (2) created subscription confirmed >



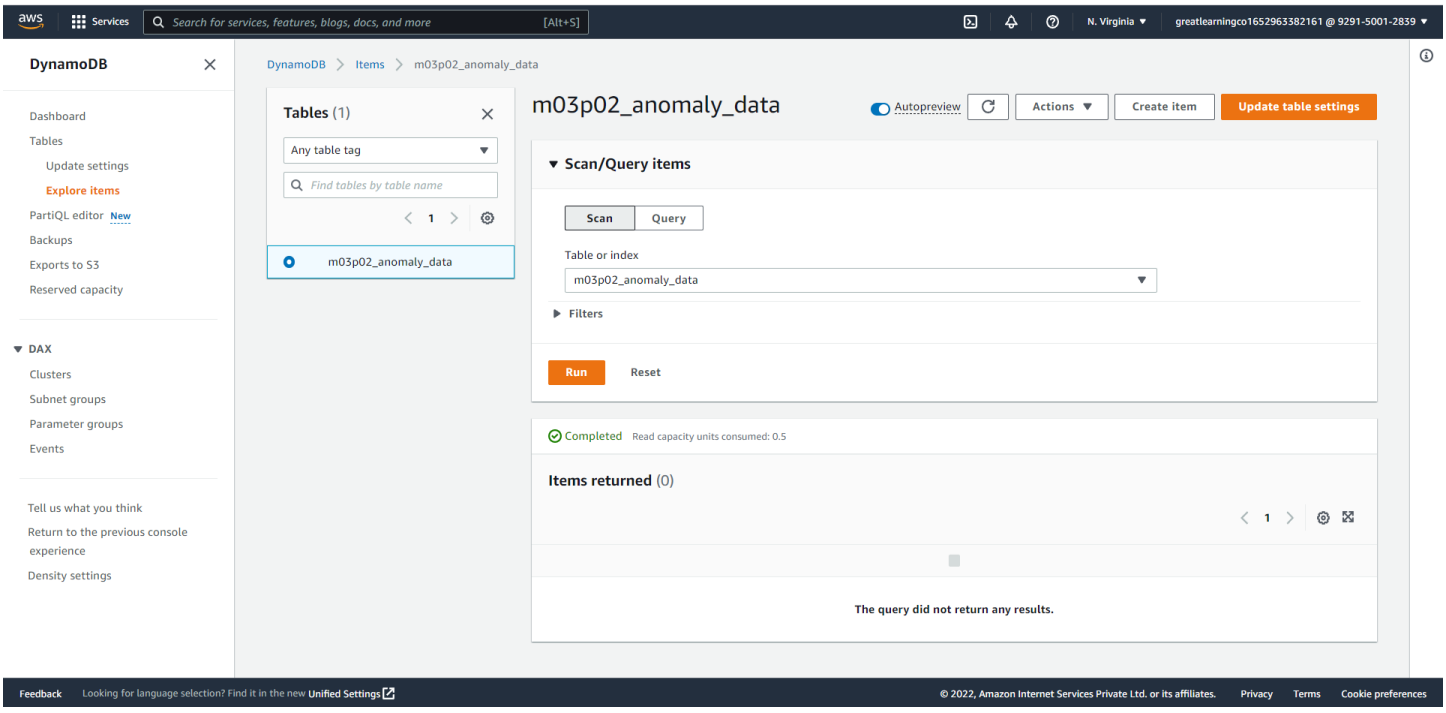
<Screenshot d (2) created subscription confirmed >



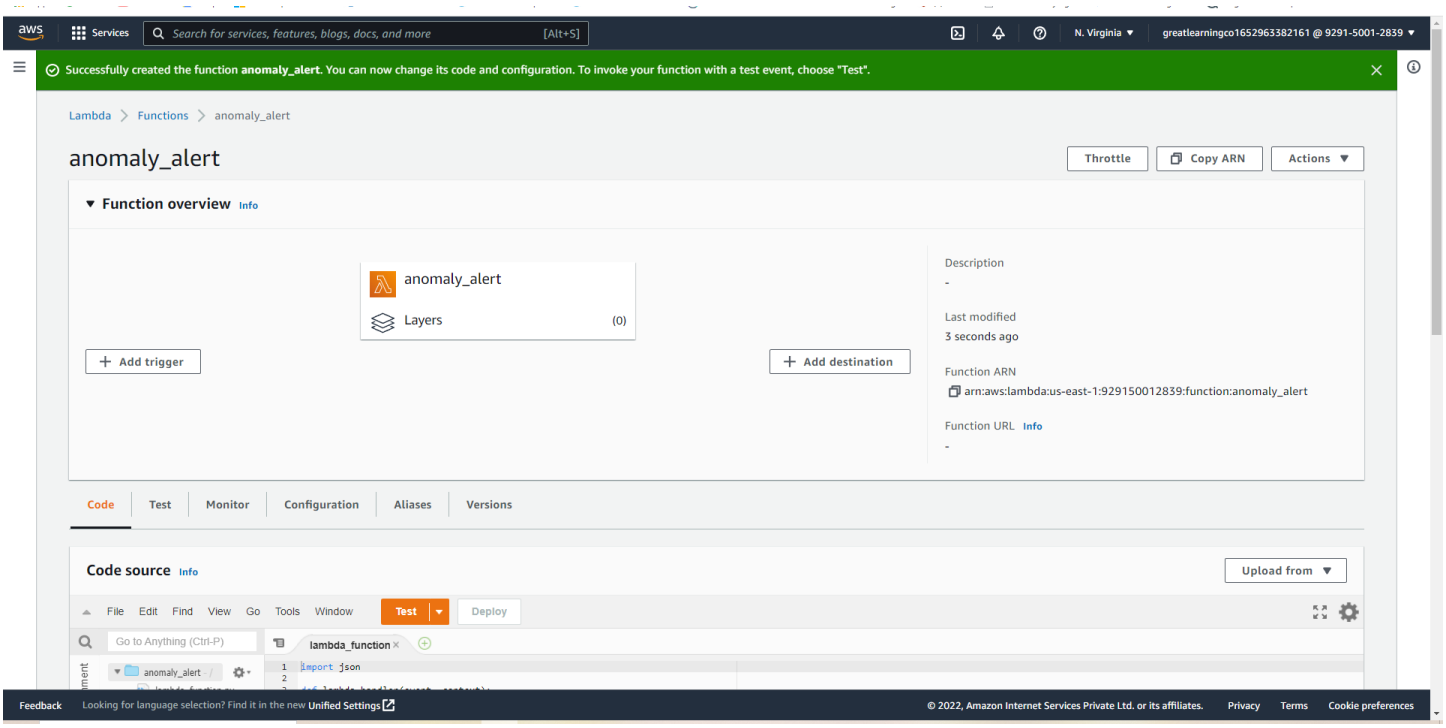
<Screenshot of created Dynamodb table----“m03p02_anomaly_data”>



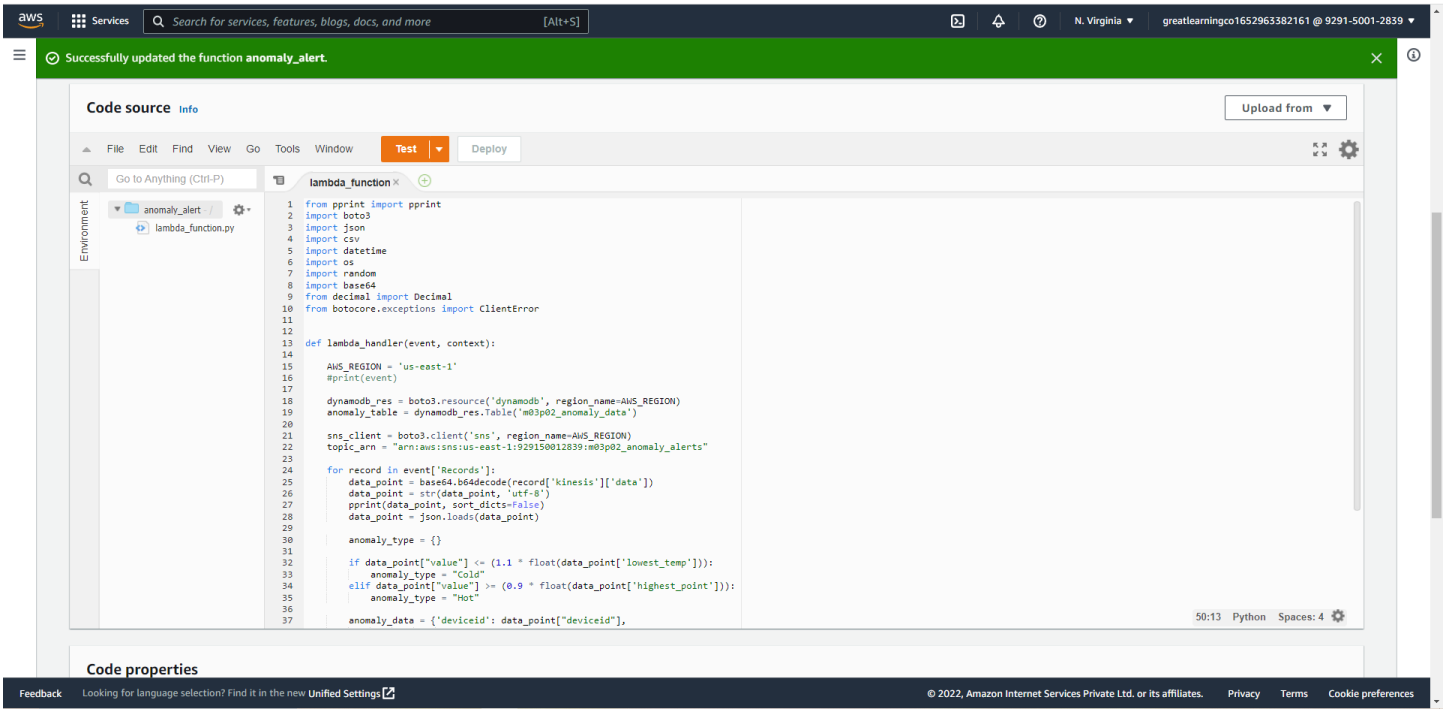
<Screenshot of created Dynamodb table----“m03p02_anomaly_data” EMPTY>



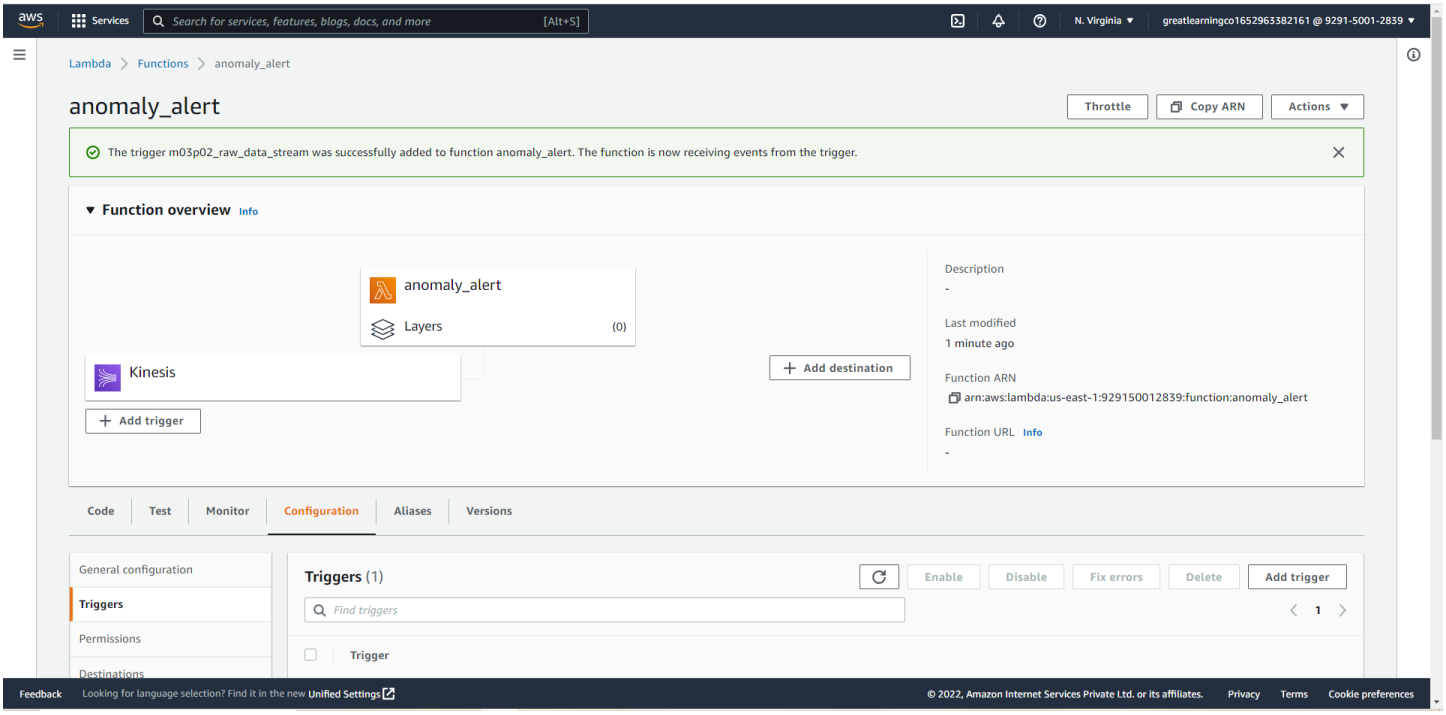
<Lambda created manually for TASK 1 >



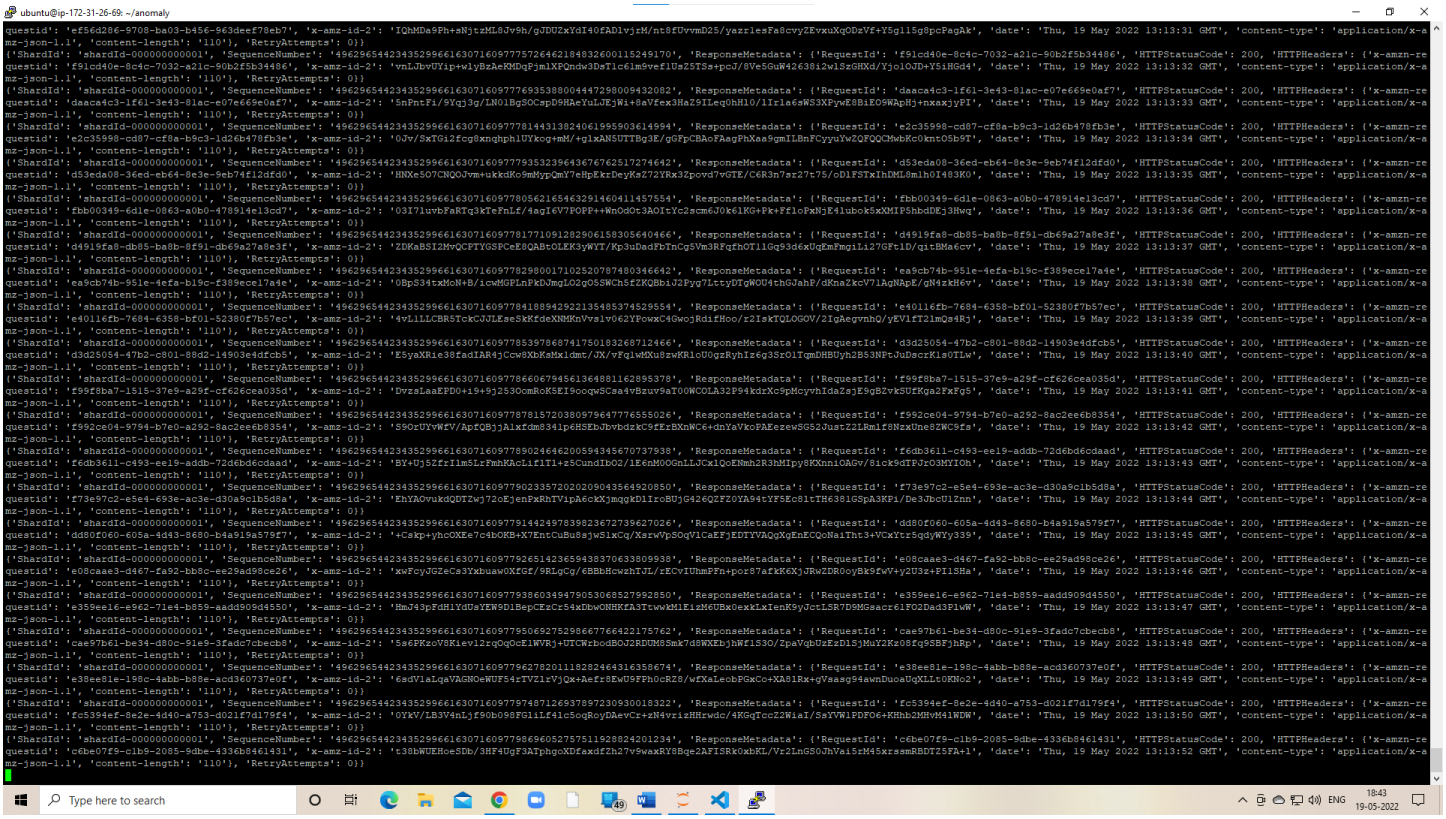
<Lambda code deployed manually for TASK 1 >



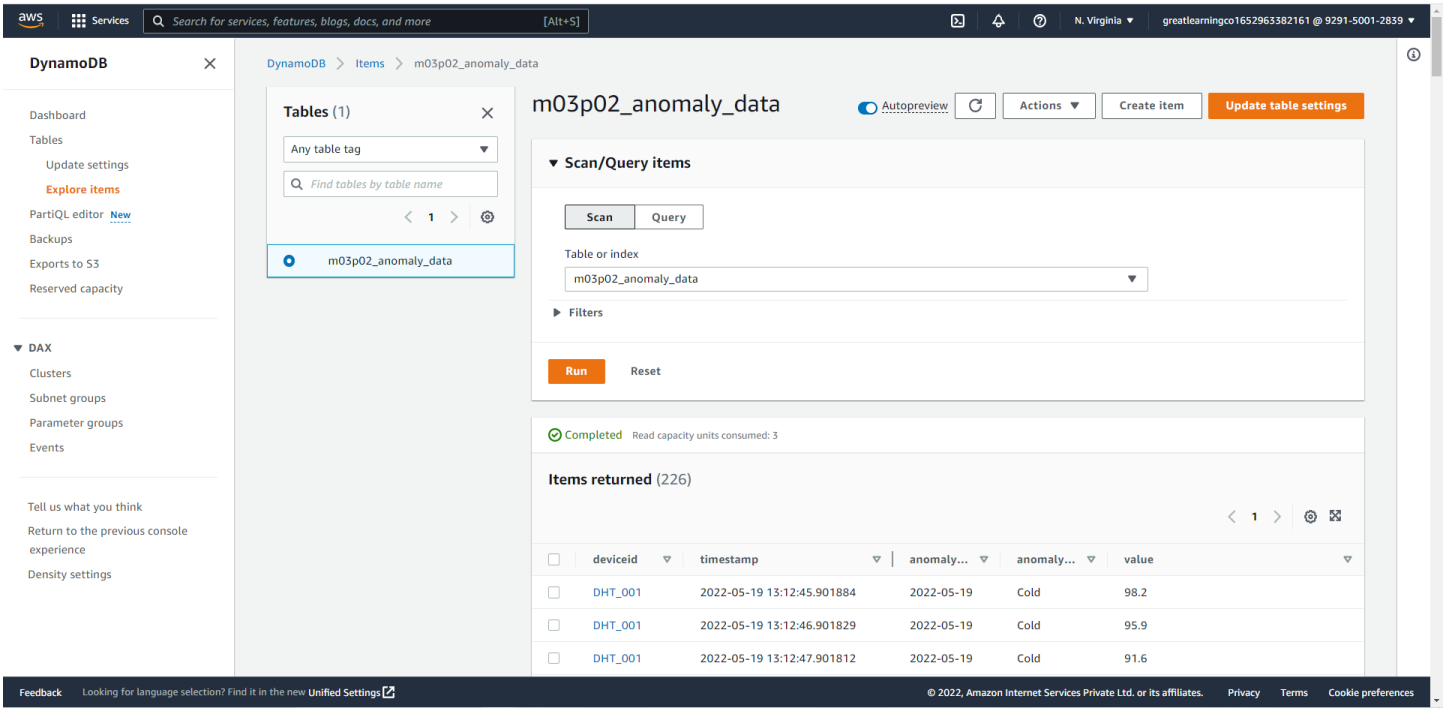
<Lambda trigger added (Kinesis data stream) manually for TASK 1 >



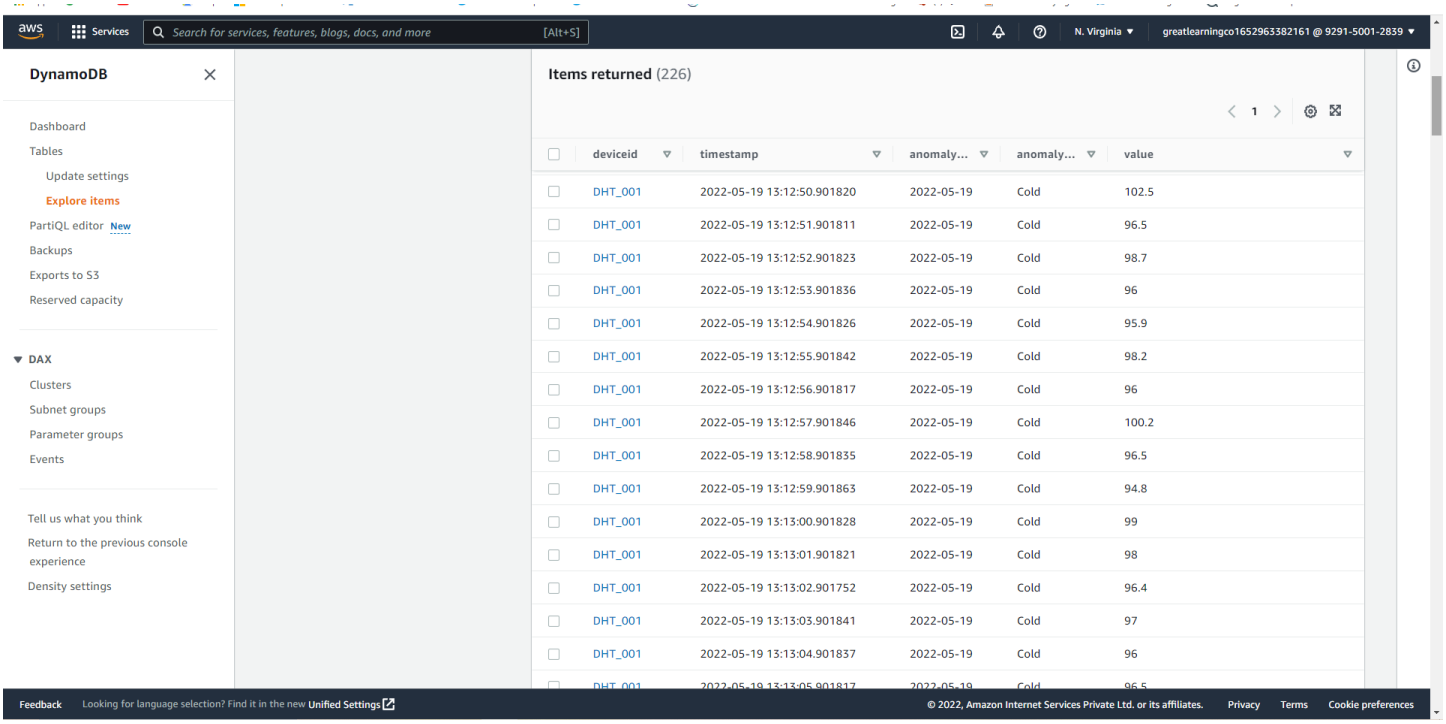
<Data push began from EC2 server (Ubuntu server) manually for TASK 1 >



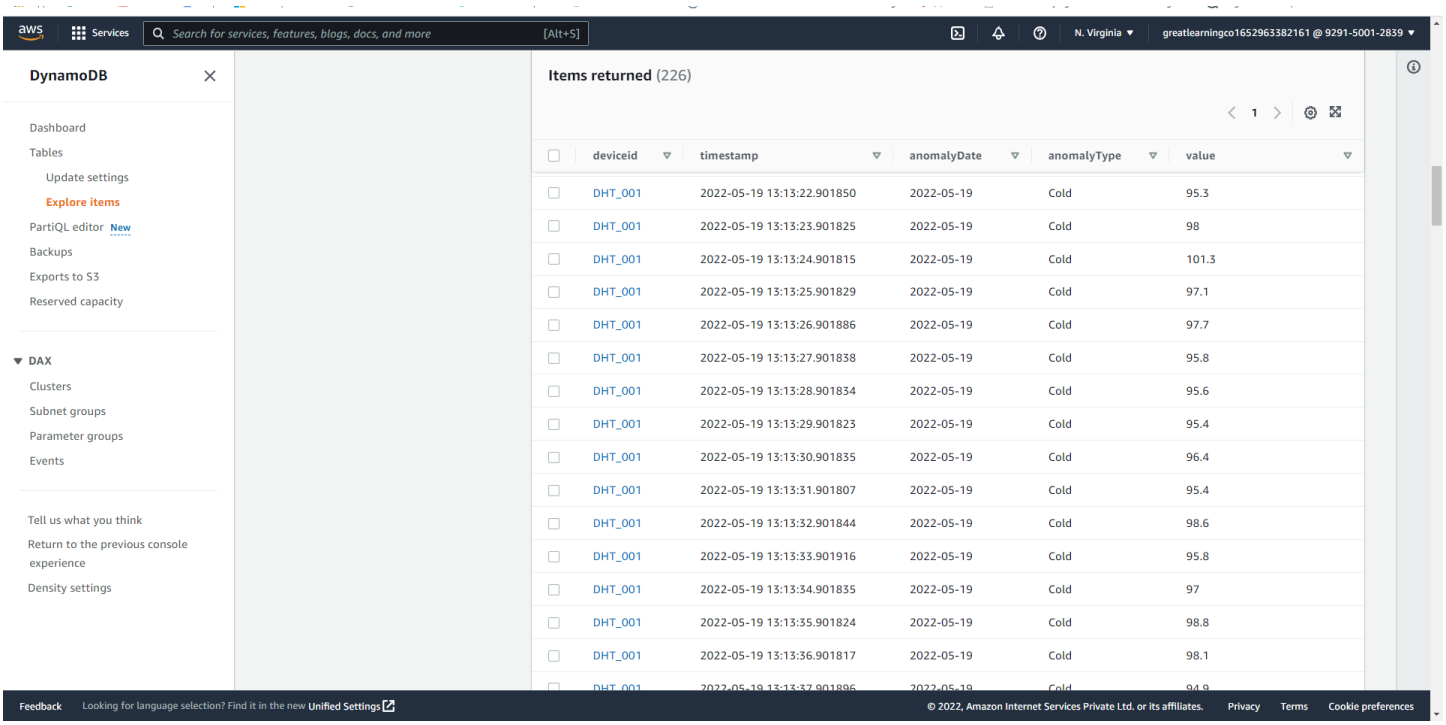
<Data POPULATED in Dynamodb table “m03p02_anomaly_data” for TASK 1 >



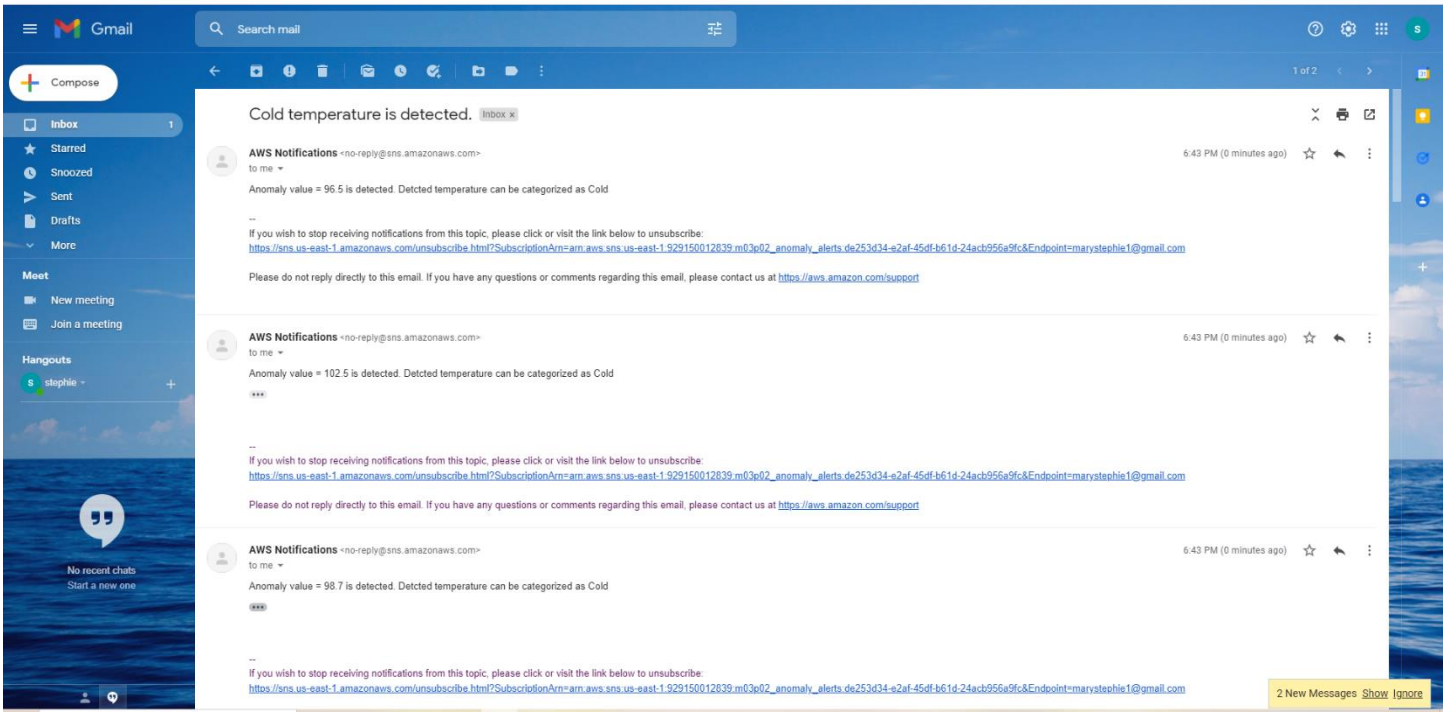
<Data POPULATED in Dynamodb table “m03p02_anomaly_data” for TASK 1 >



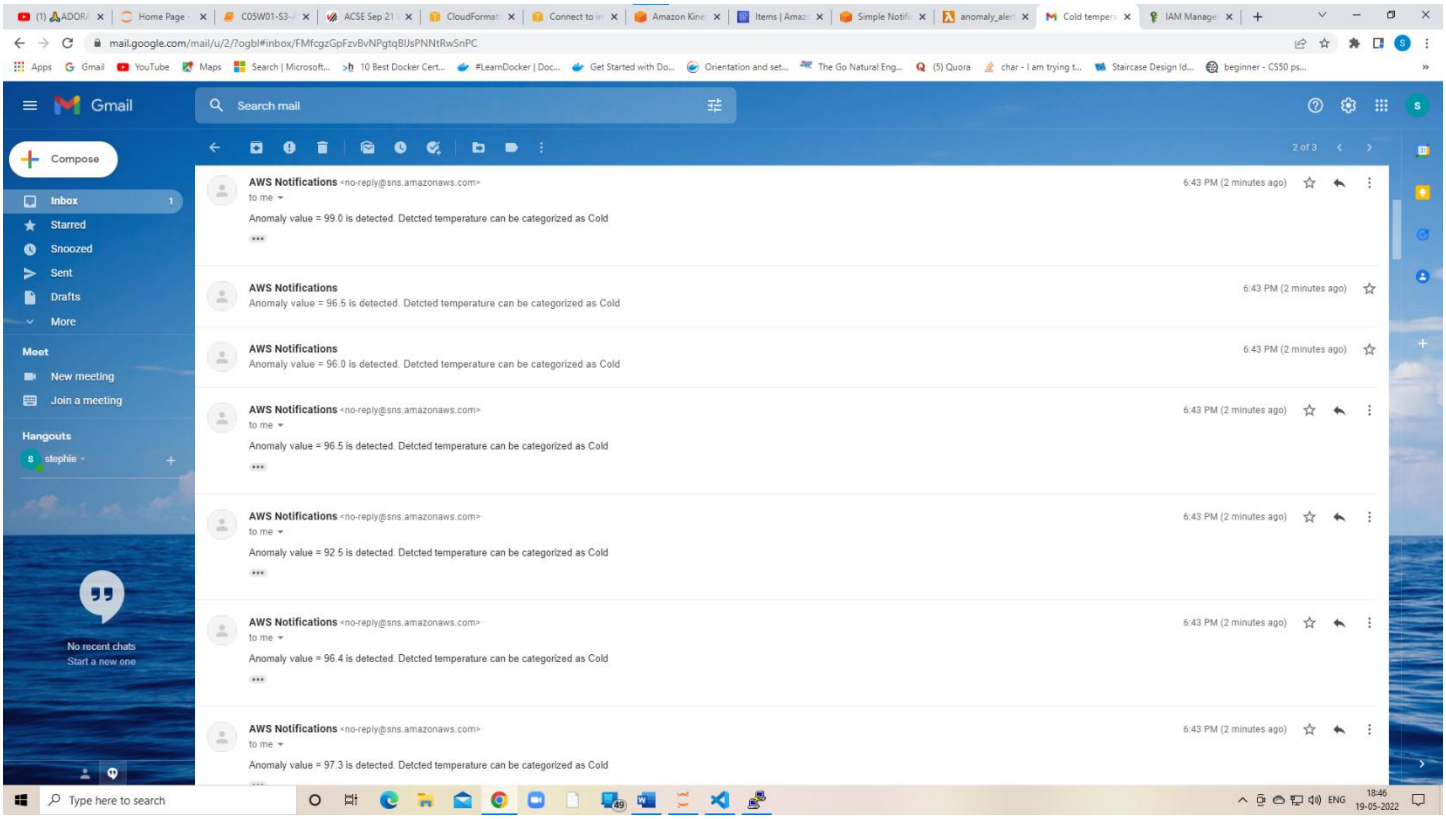
<Data POPULATED in Dynamodb table “m03p02_anomaly_data” for TASK 1 >



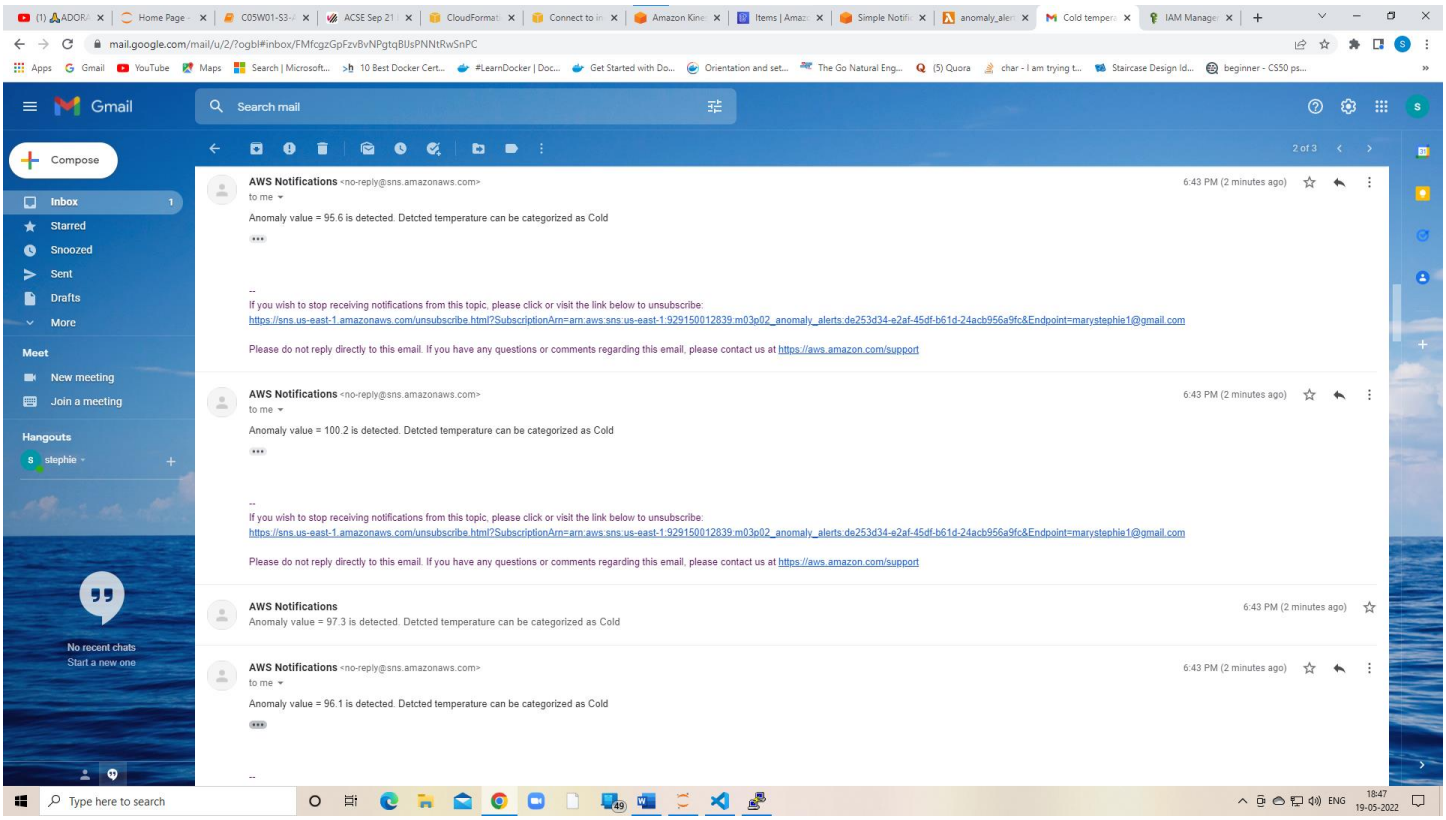
<SNS notification via email received for TASK 1 >



<SNS notification via email received for TASK 1 >



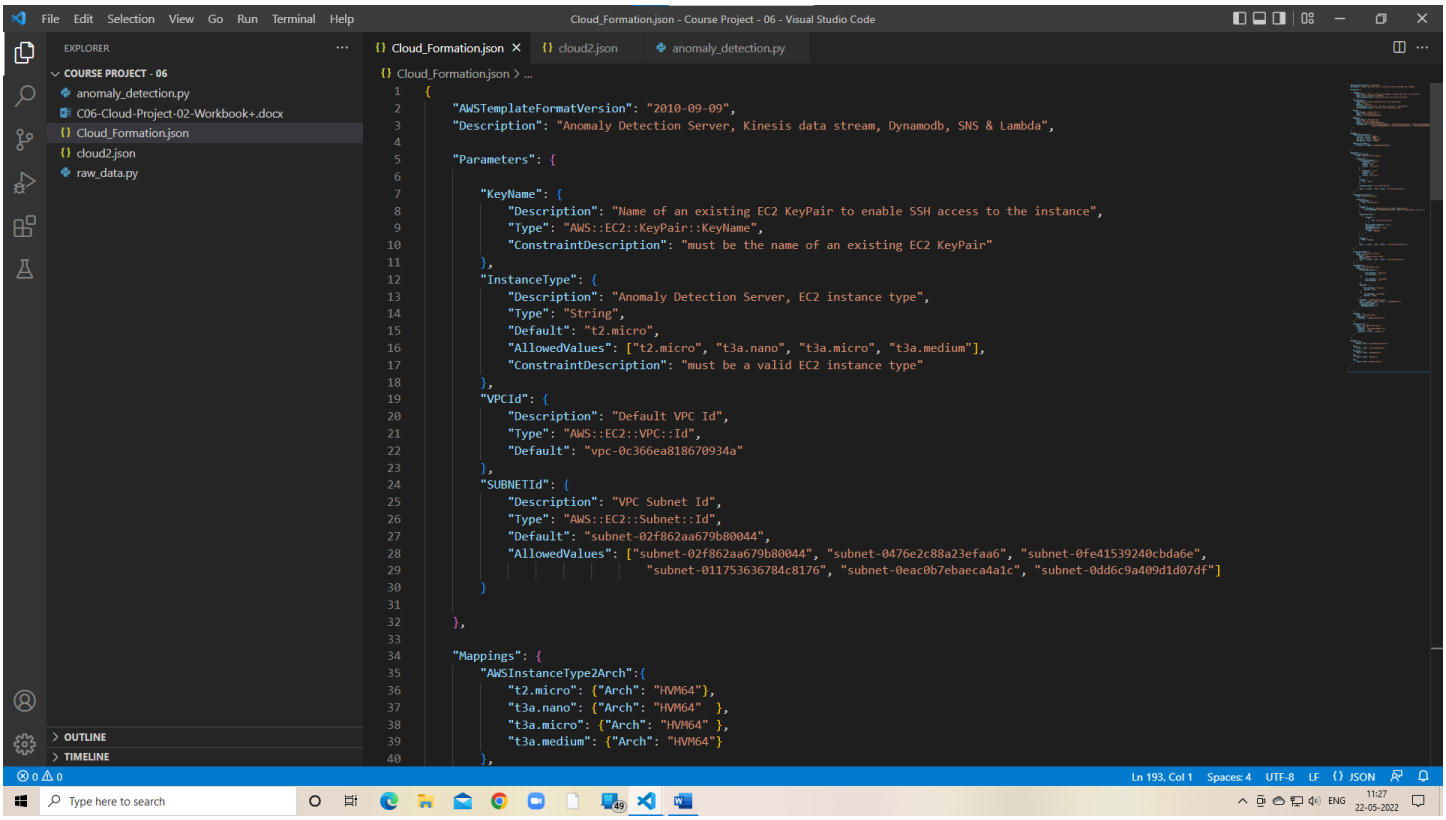
<SNS notification via email received for TASK 1 >



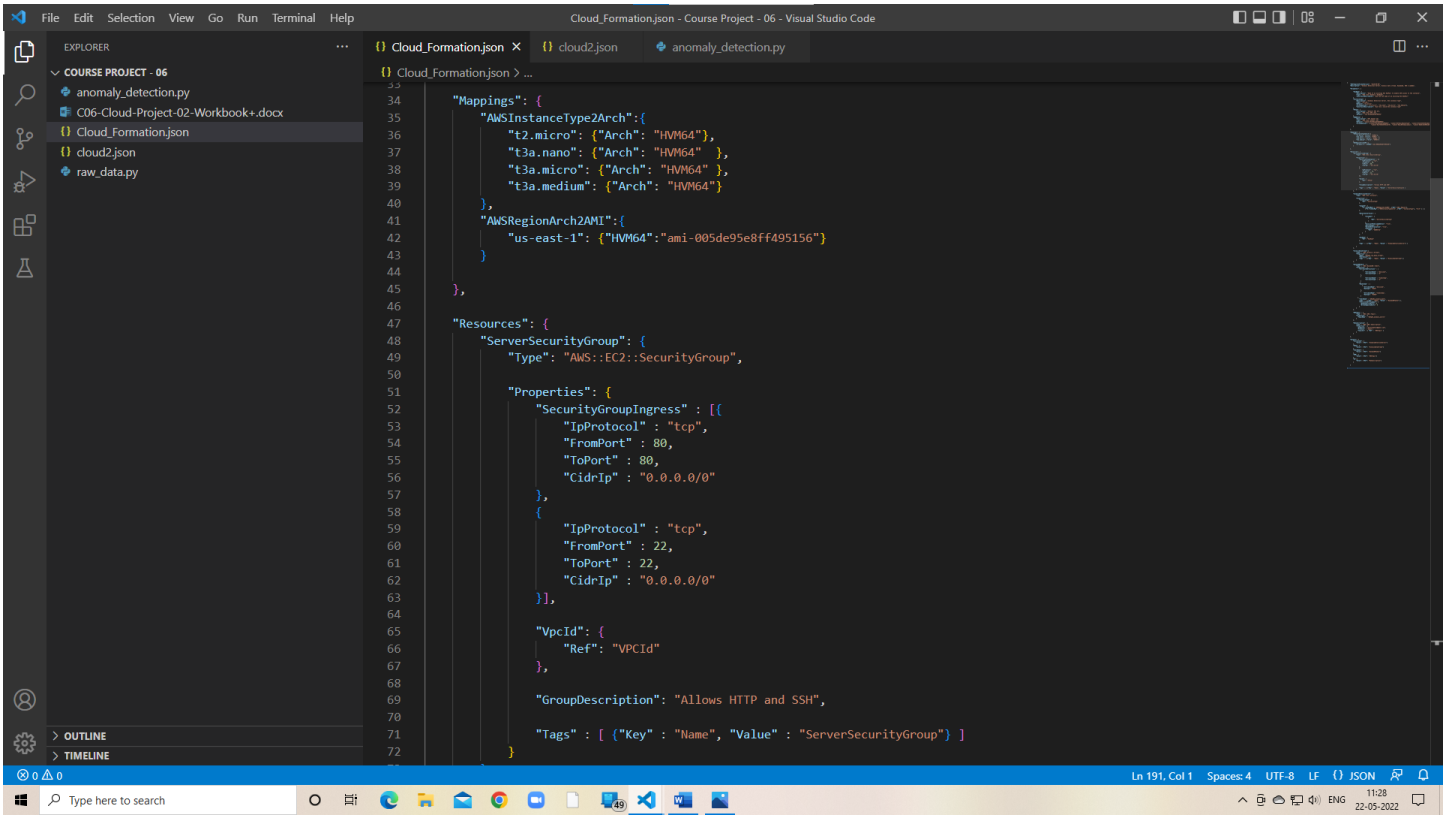
Task 2 Screenshots:

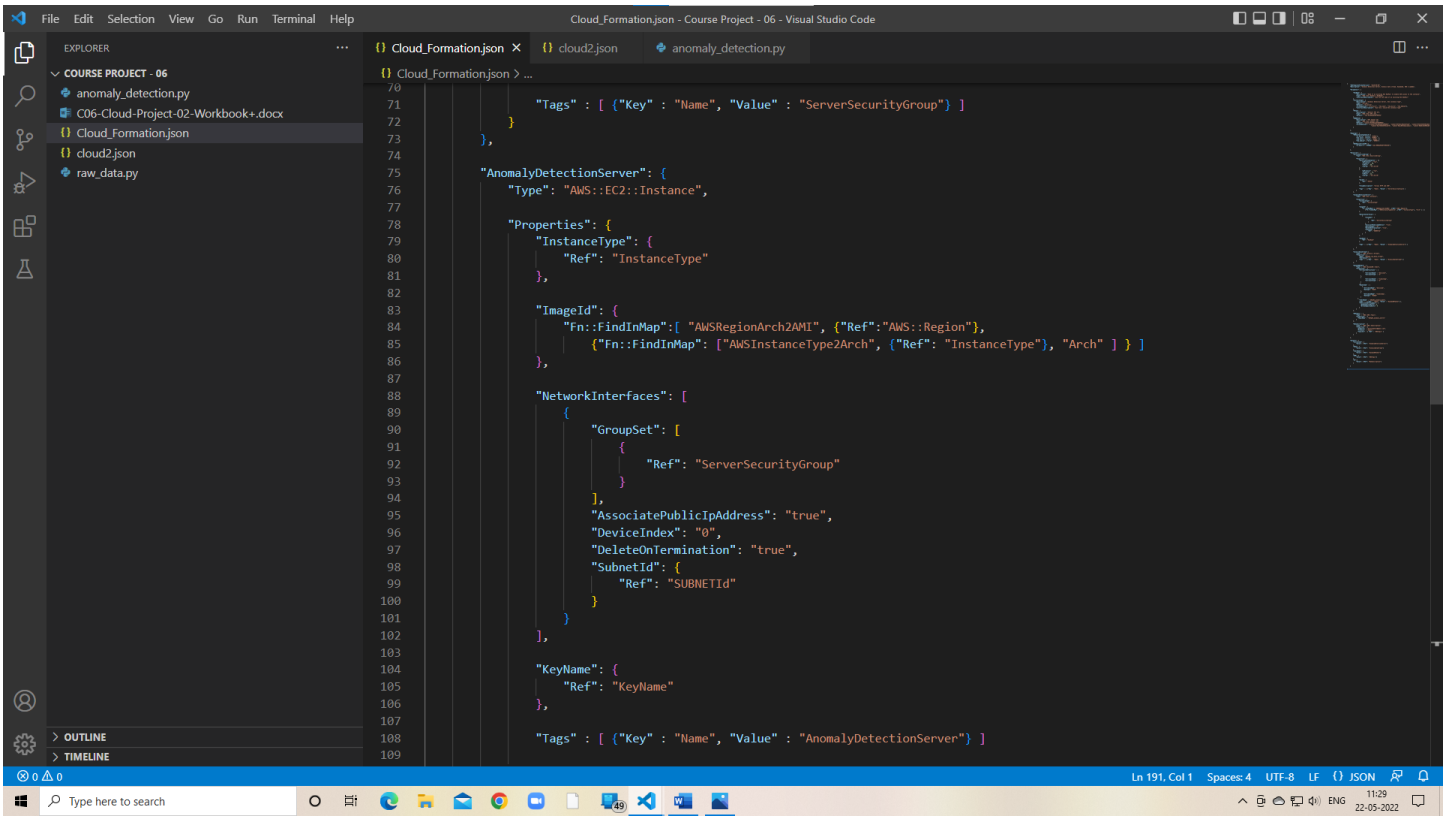
Step number	a
Step name	Cloud formation summary page
Instructions	1) Template file (JSON script) to enable mentioned services in Task -1 and Task-2
Expected screenshots	1) Cloud formation summary page

<Screenshot for Cloud_Formation.json template for TASK 2>

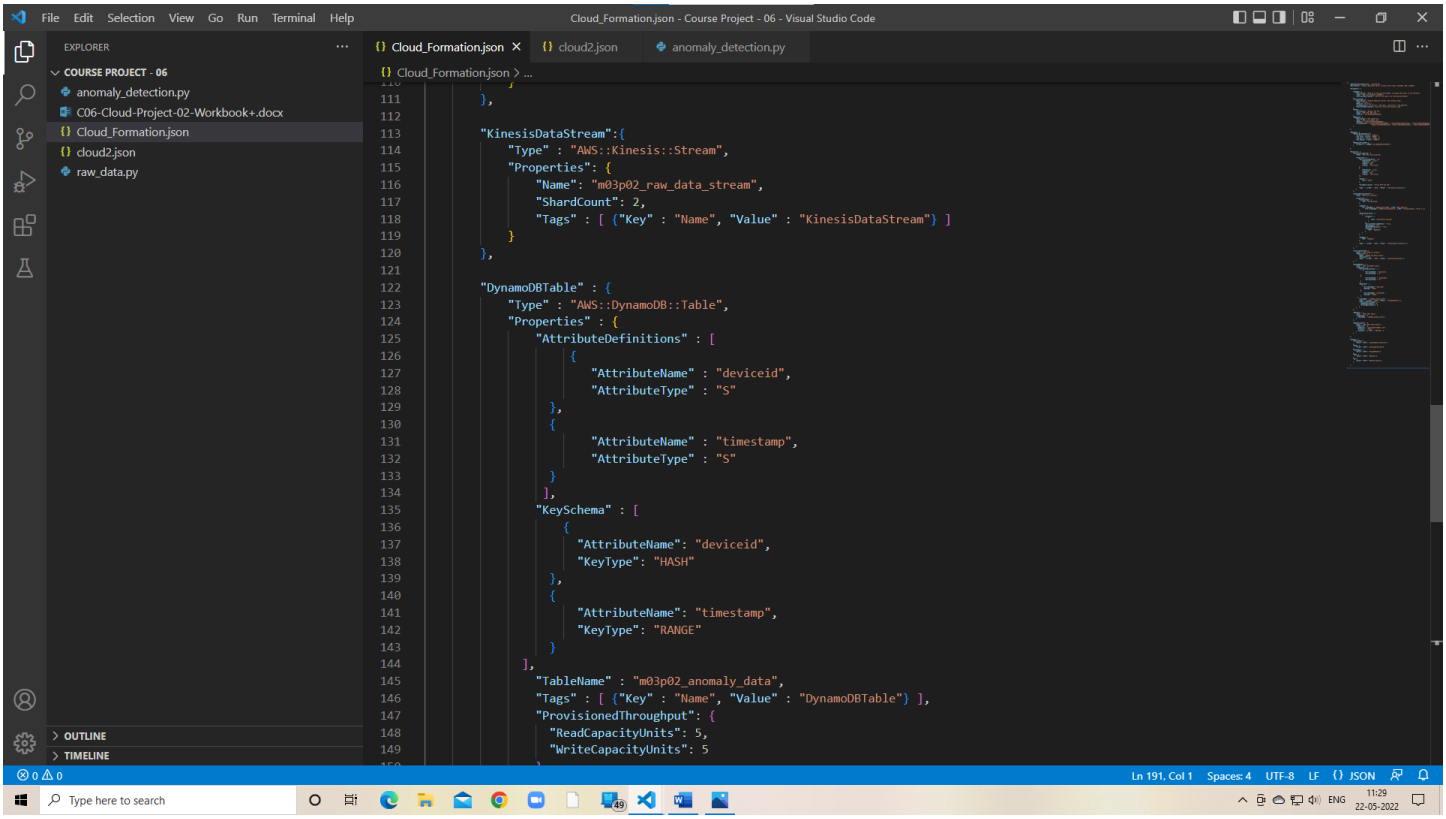


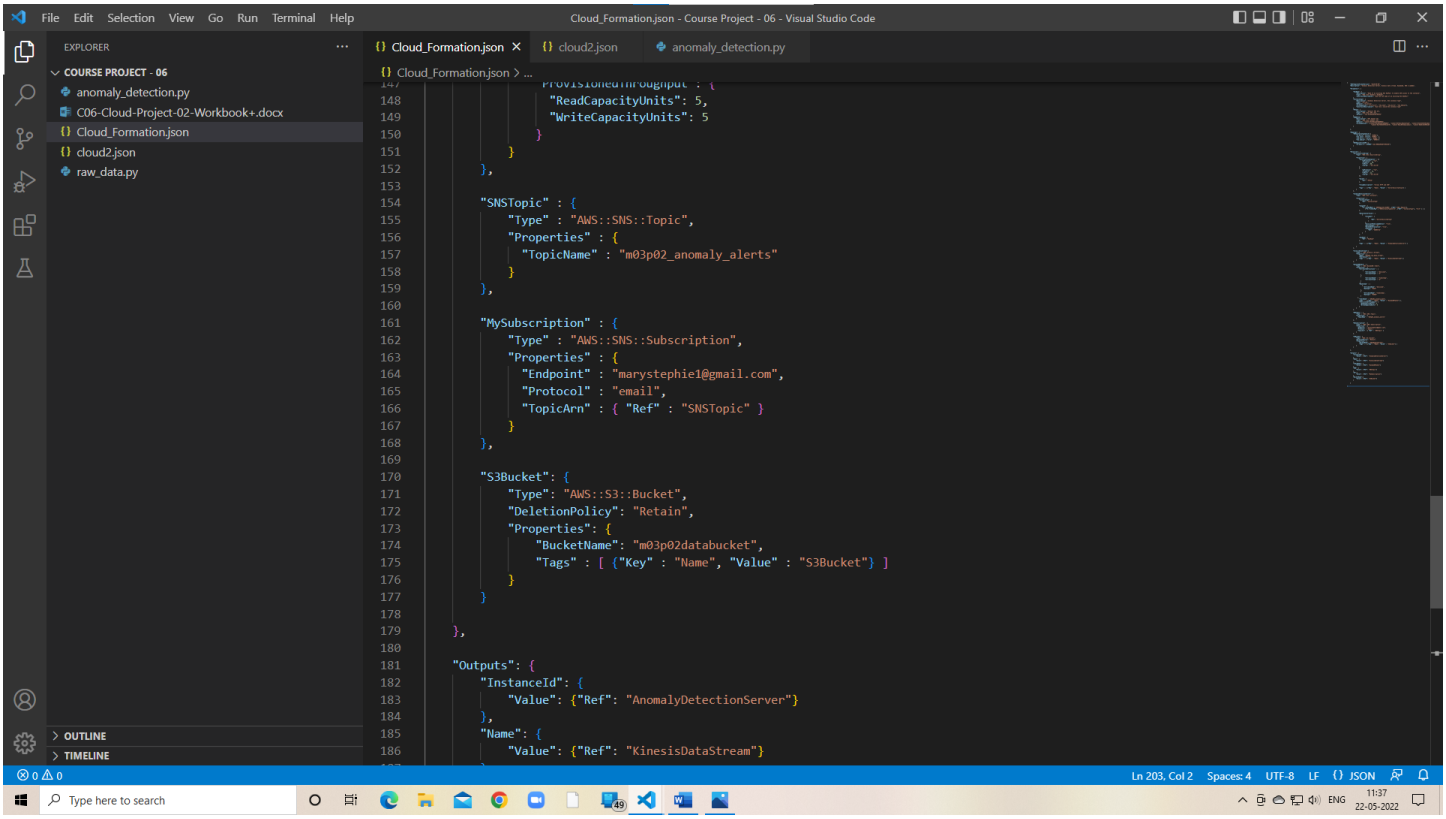
<Screenshot for Cloud_Formation.json template for TASK 2>

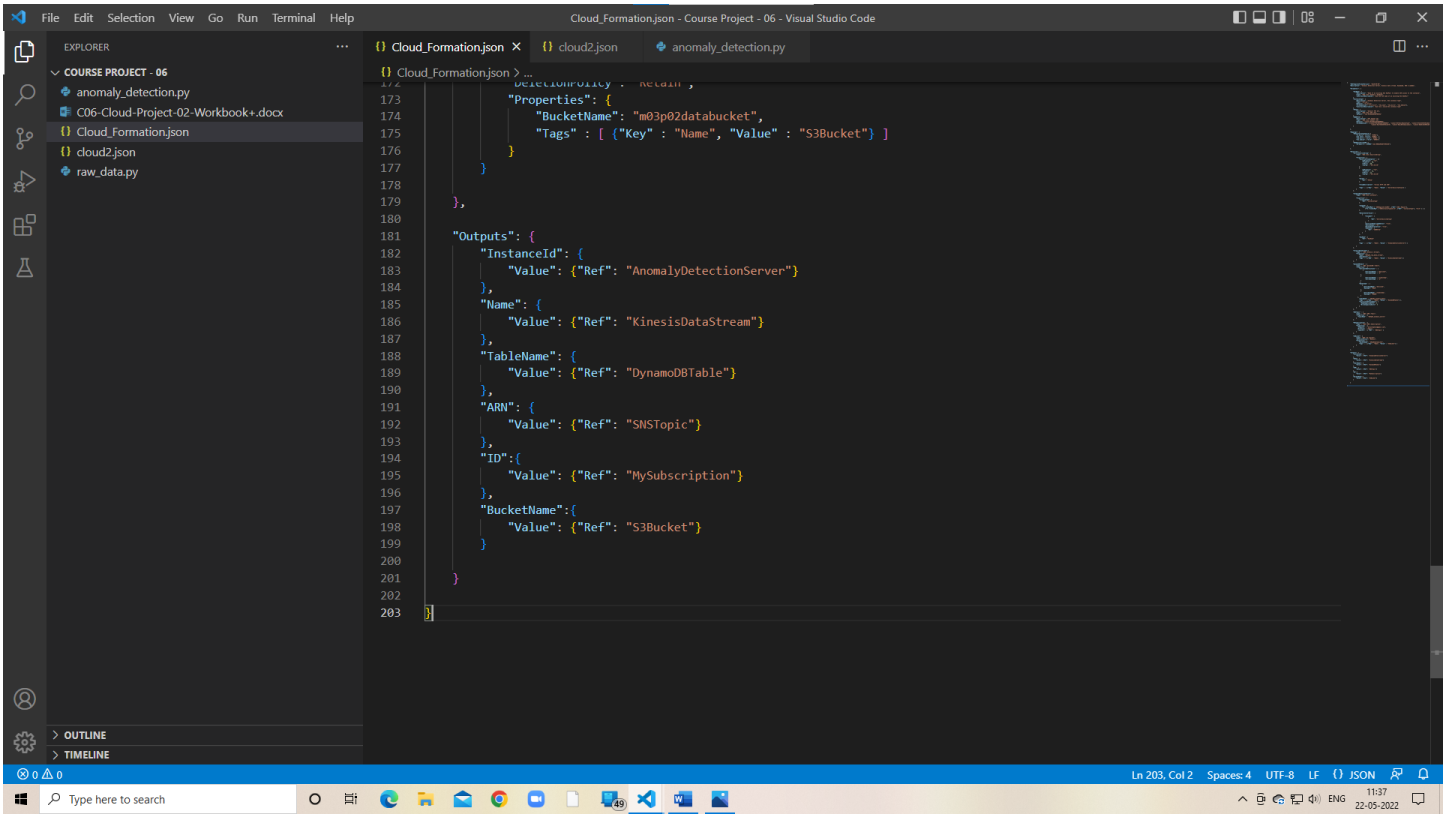




<Screenshot for Cloud_Formation.json template for TASK 2>

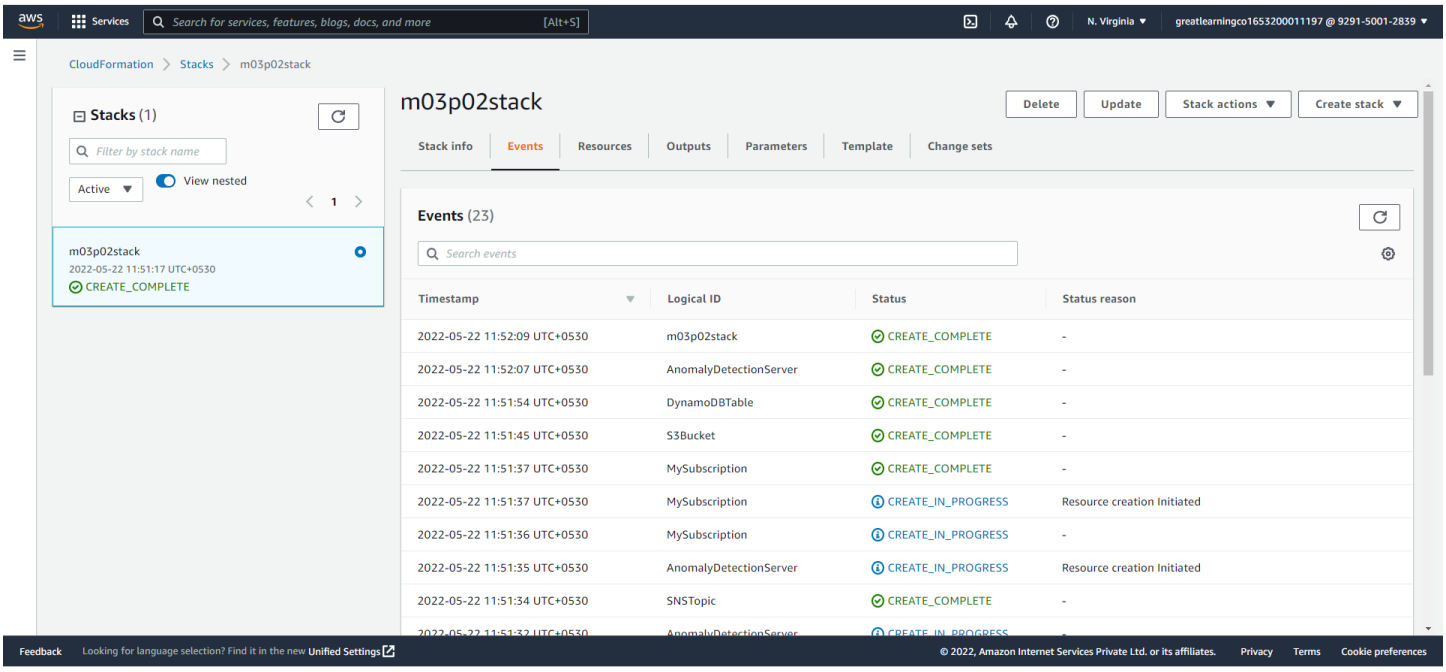




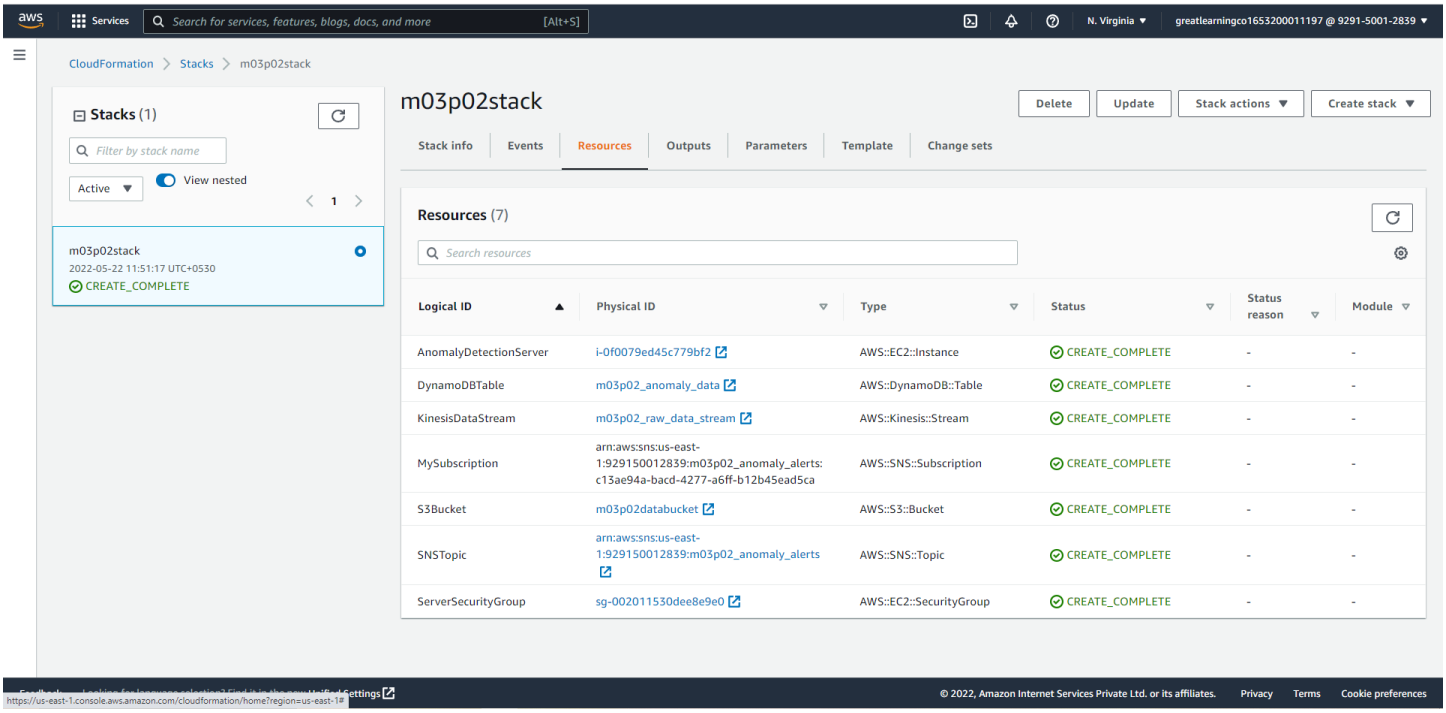


< Screenshot a (1) Cloud Formation stack successfully created>

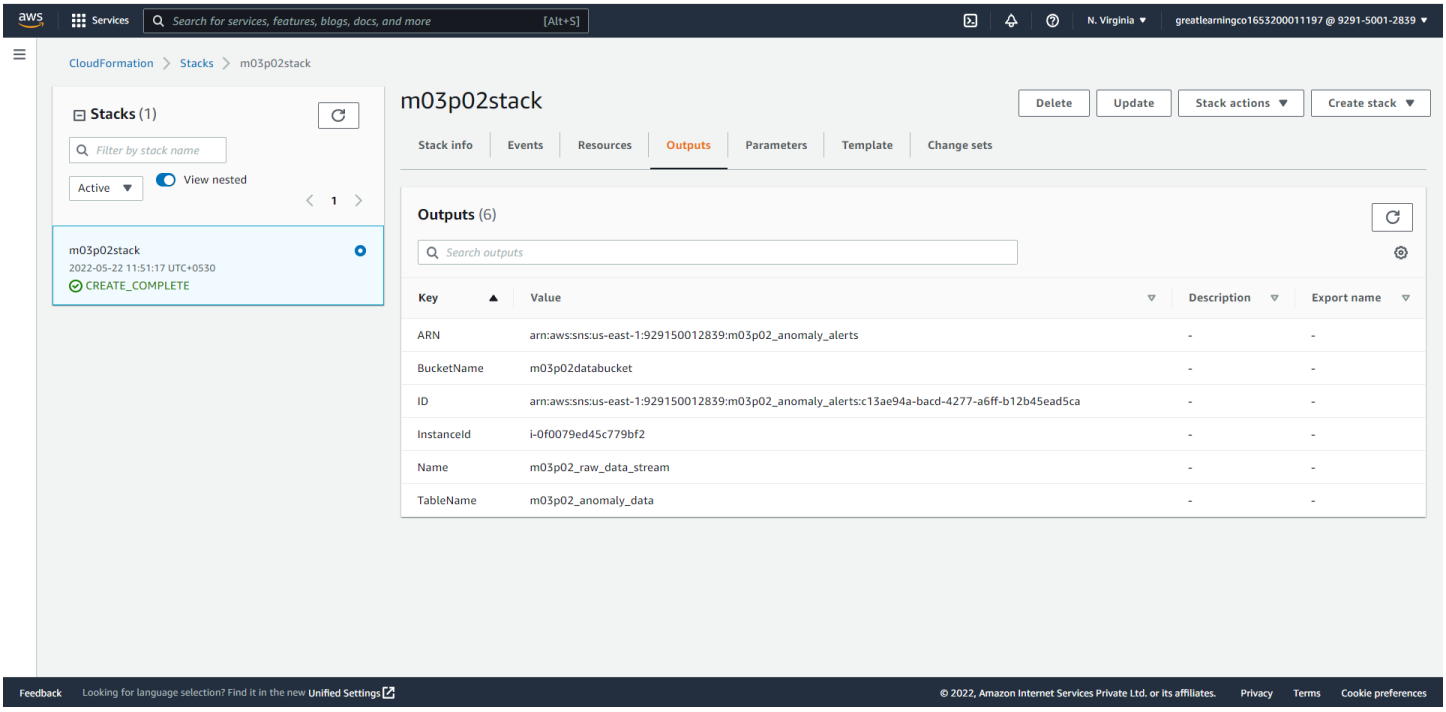
<Screenshot of Cloud Formation “m03p02stack” with s3 bucket service added for TASK 2>



<Screenshot of “m03p02stack” successful “Resources” for TASK 2>



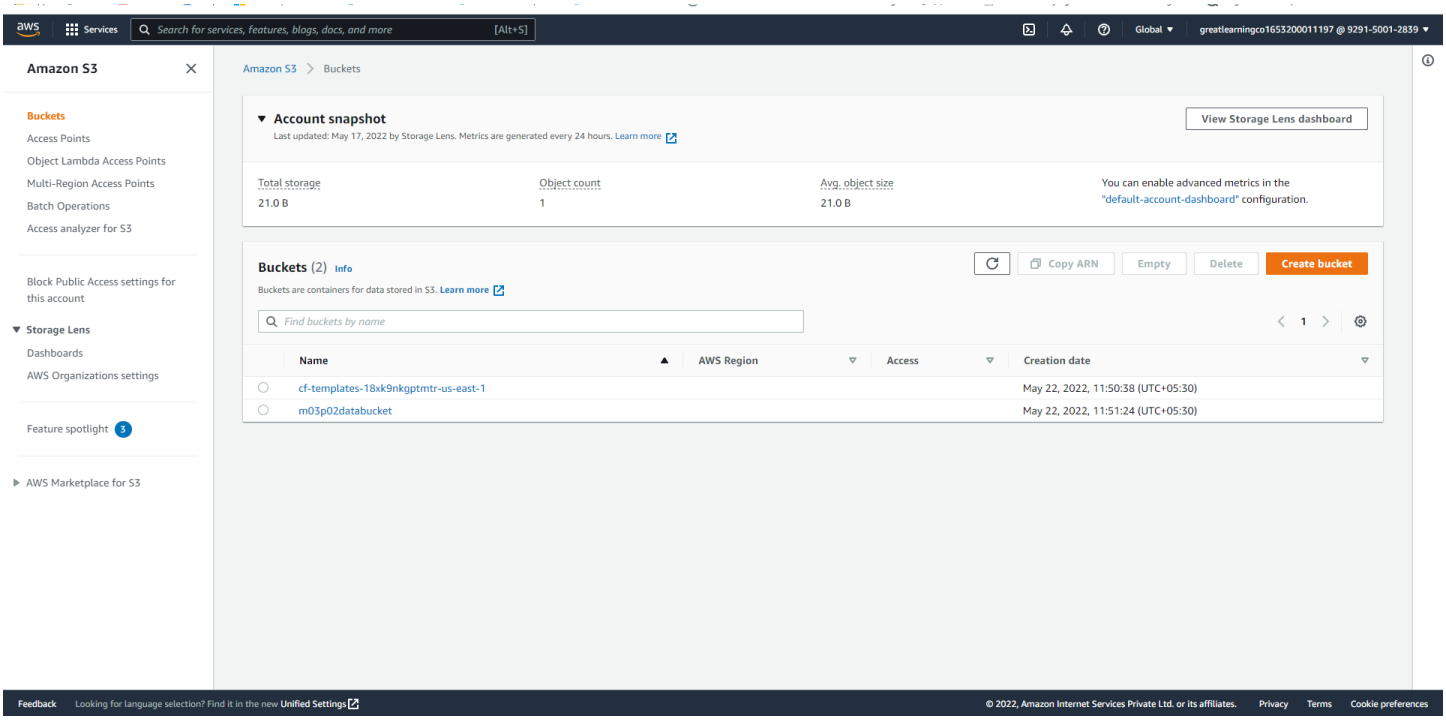
<Screenshot “m03p02stack” successful “Outputs” for TASK 2>



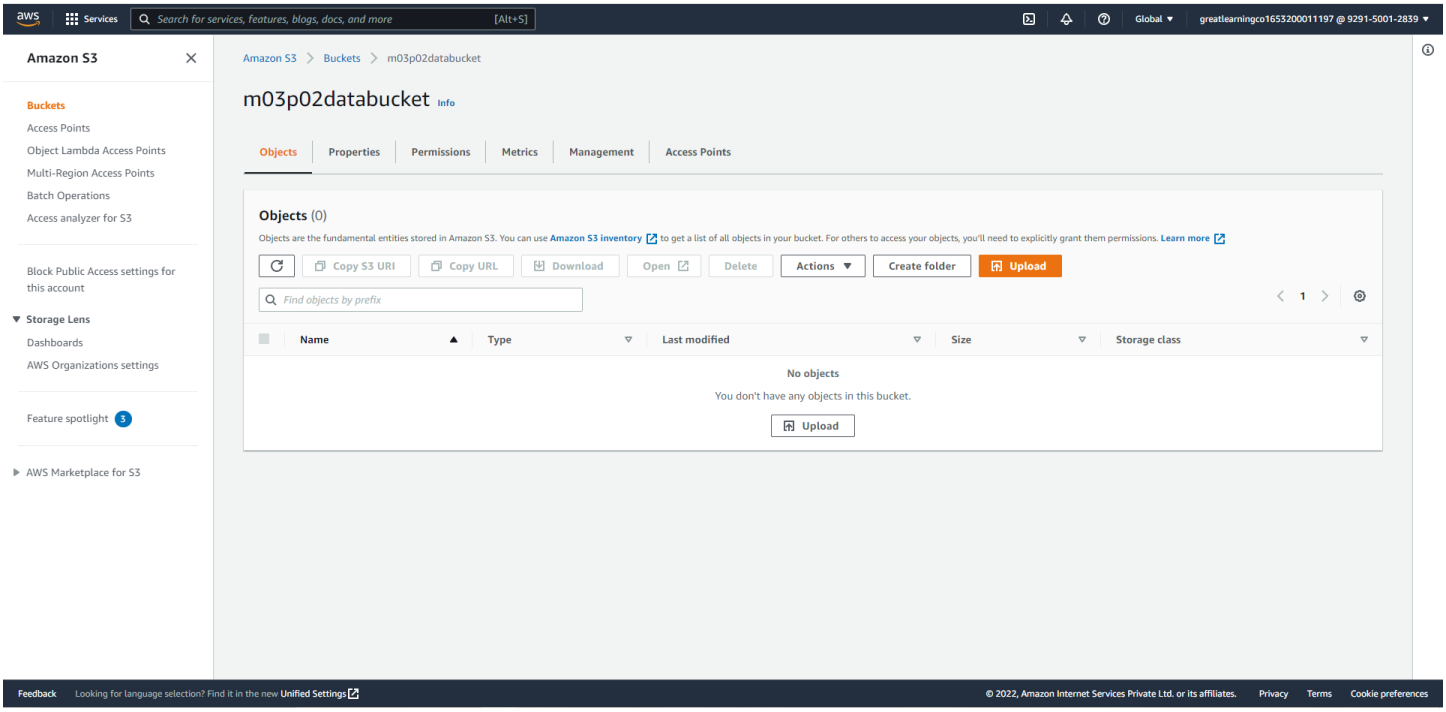
Step number	b
Step name	Enabled S3 service
Instructions	1) Make sure that cloud formation operation is successful 2) Check if the S3 service are successful, if yes, goto S3 console and assure the created directory is available 3) Upload the relevant scripts and appsec.yml
Expected screenshots	1) Enabled empty S3 service 2) S3 directory after uploading the application 3) S3 directory after uploading appsec.yml

<Screenshots for b (1) >

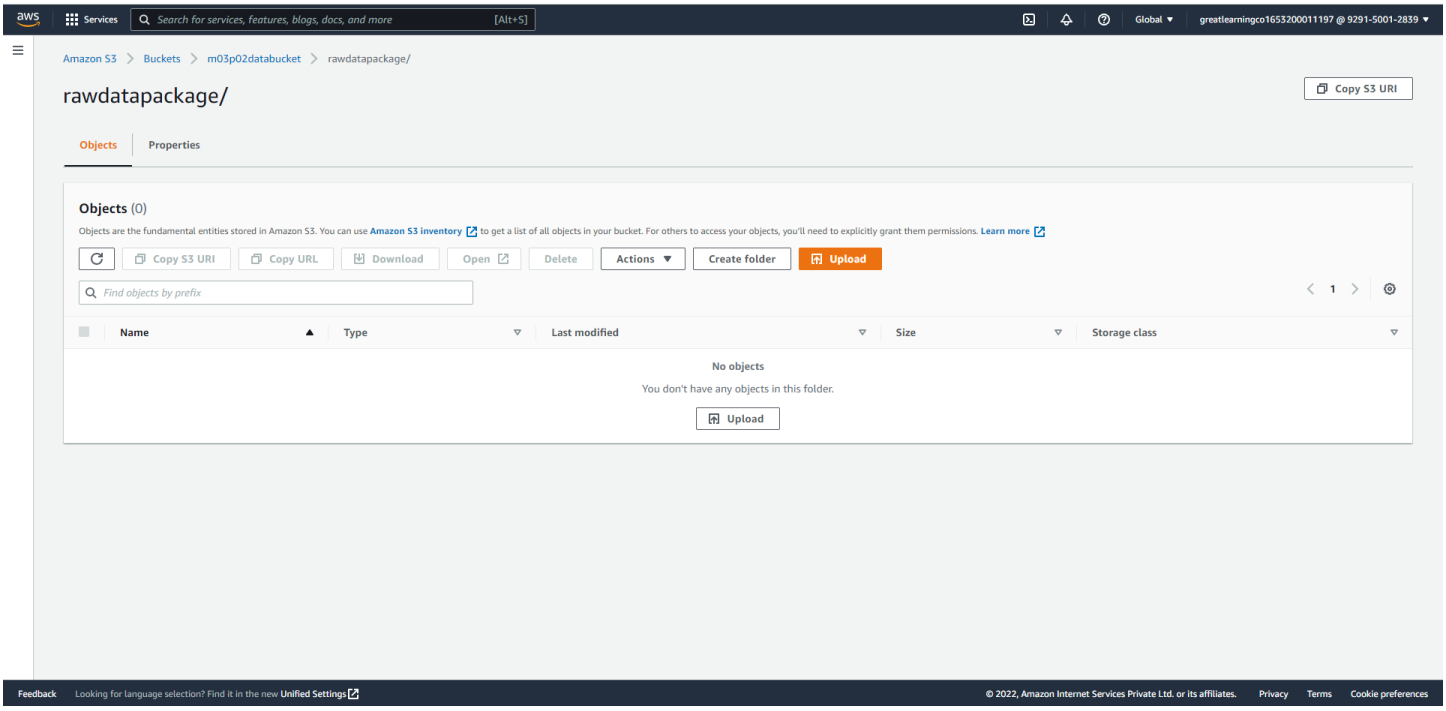
<Screenshot of Enabled S3 Service, “m03p02databucket” successfully created for TASK 2>



<Screenshot “m03p02bucket” empty for TASK 2>

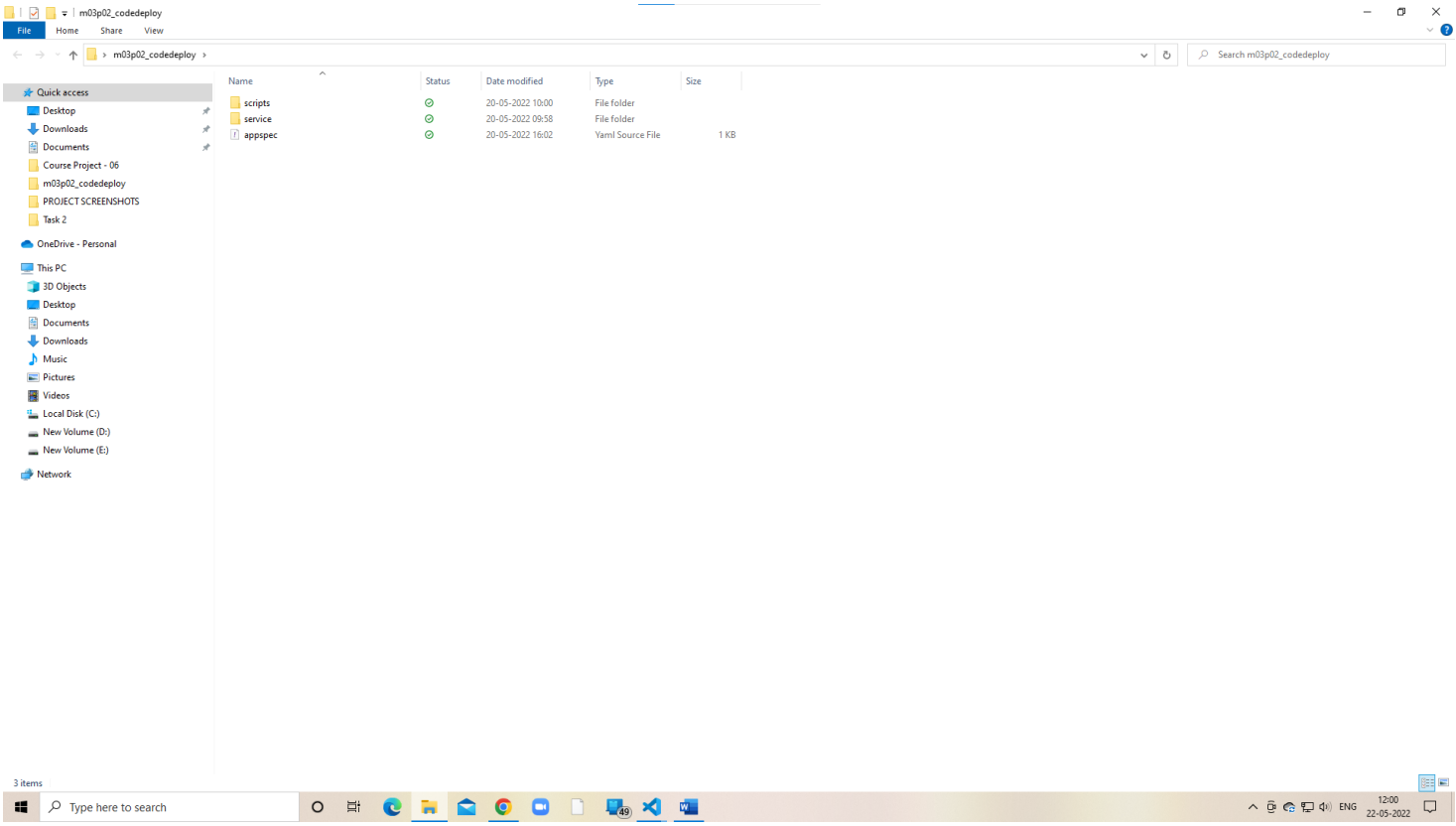


< Screenshot of first empty folder where.zip file will be uploaded for TASK 2 >

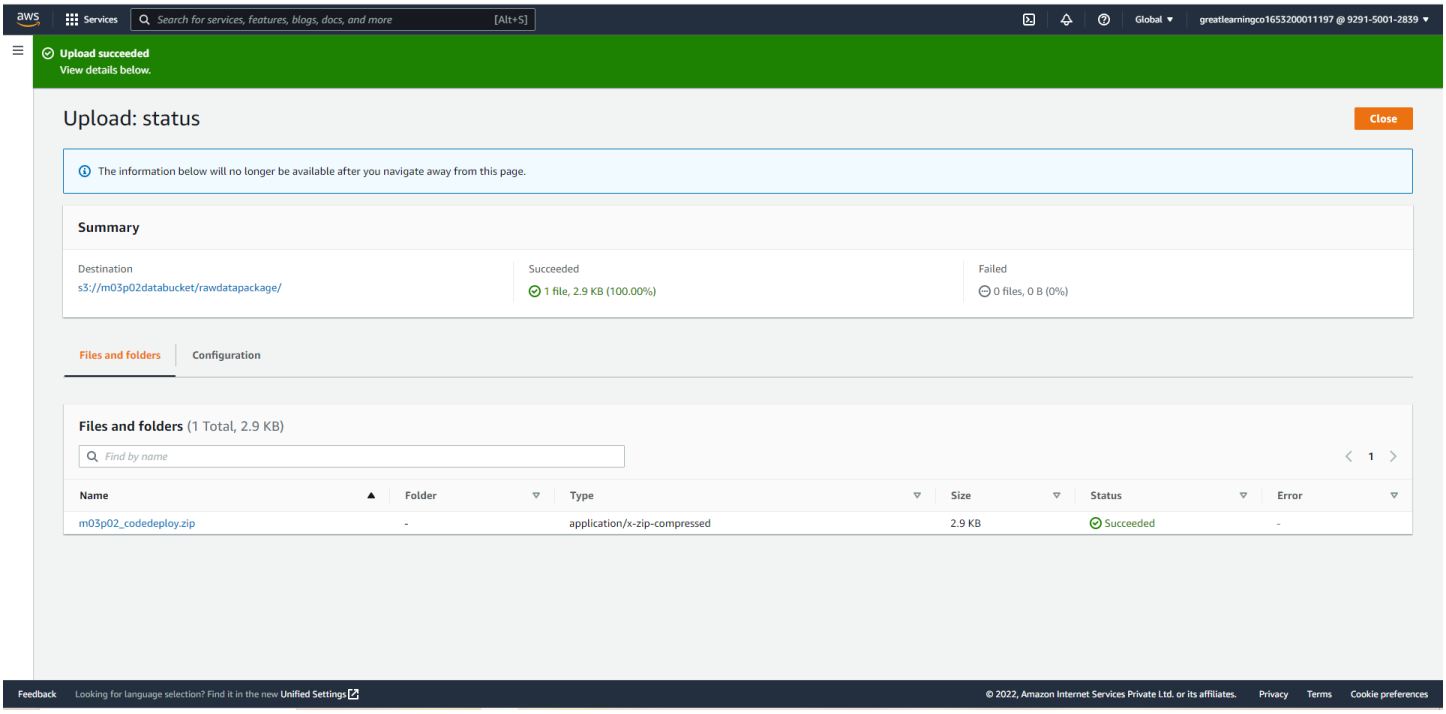


<Screenshots for b (2) and b (3) >

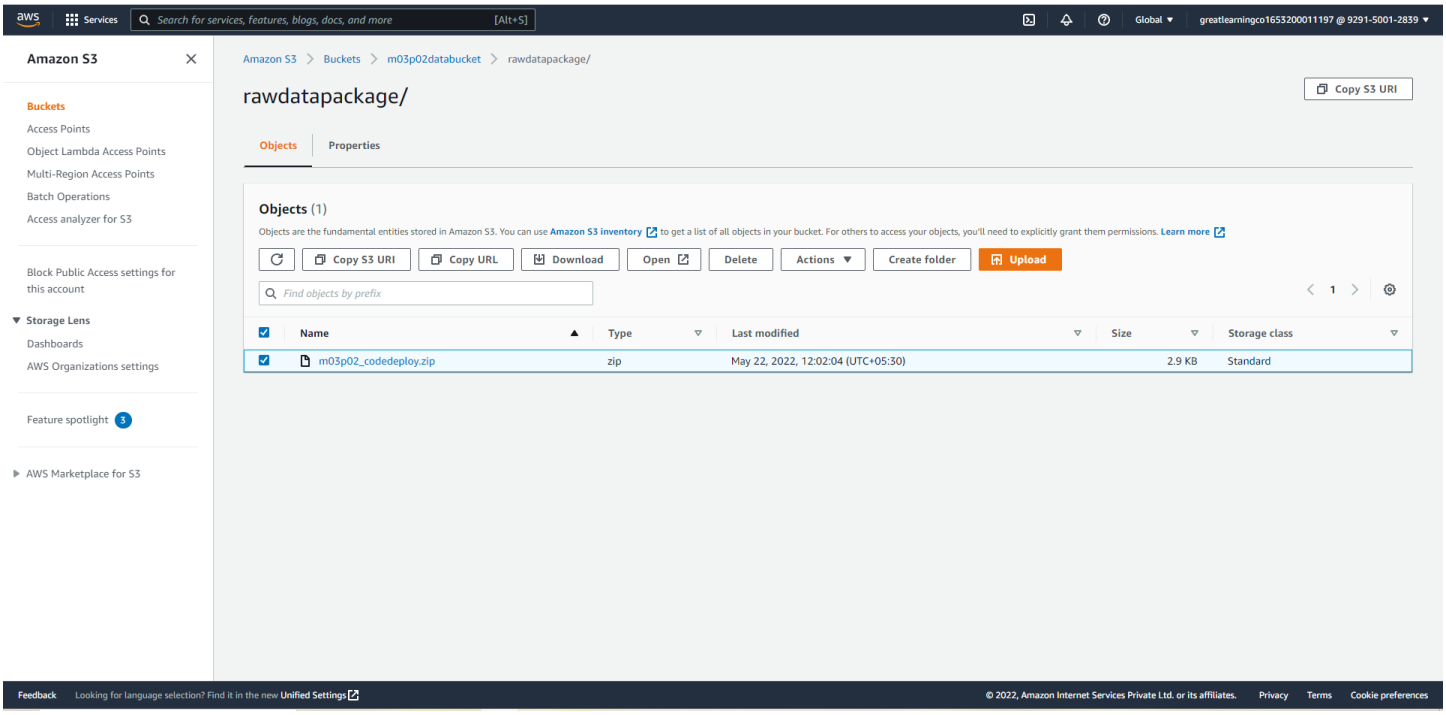
< Screenshot of “m03p02_codedeploy.zip” folder which needs to be uploaded in S3 bucket for TASK 2>



<Screenshot of “m03p02_codedeploy.zip” which contains both application and appspec.yml uploaded for TASK 2>

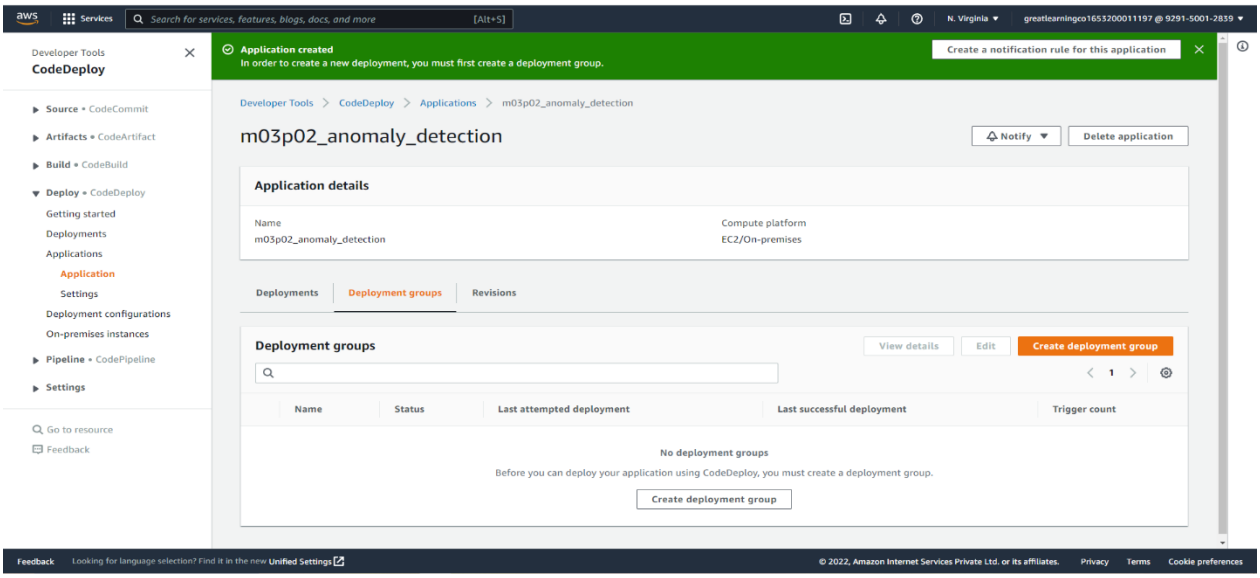


<Screenshot of “m03p02_codedeploy.zip” which contains both application and appspec.yml uploaded for TASK 2>

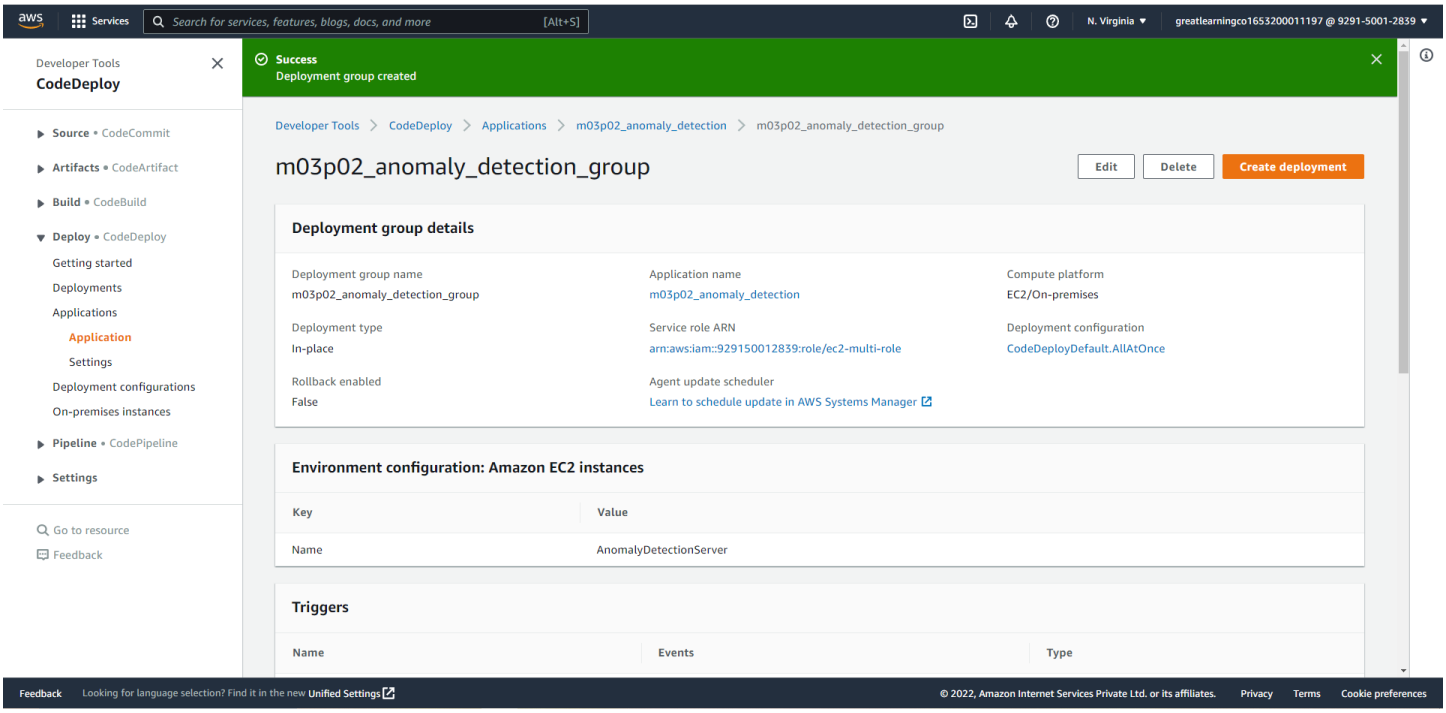


Step number	c
Step name	CodeDeploy
Instructions	1) Perform the CodeDeploy operation through AWS console 2) If CodeDeploy is successful look if the Anomaly detection related data is going into the relevant DB and SNS services
Expected screenshots	1) Successful CodeDeploy operation summary page 2) Anomaly data pushed in the DynamoDB 3) SNS notification

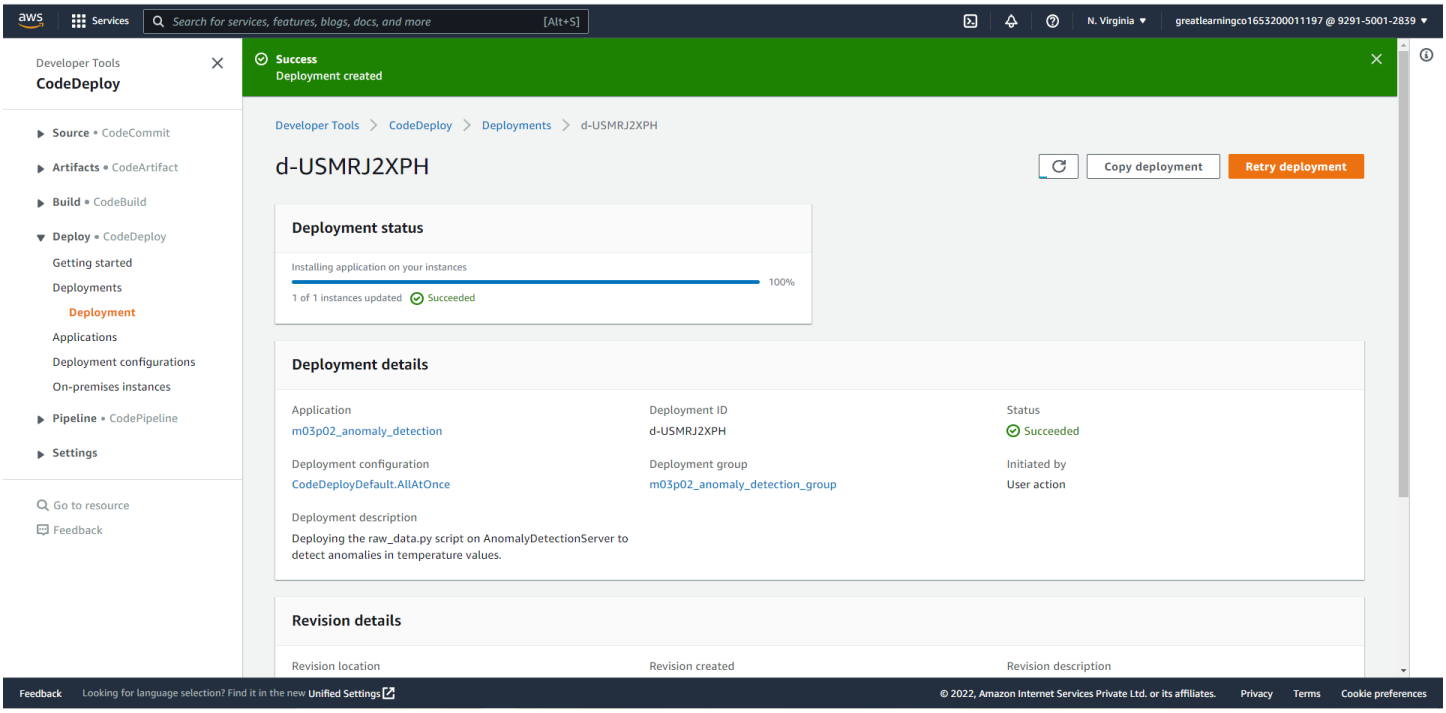
<Screenshot c (1)>
<Screenshot of CodeDeploy application creation for TASK 2 >



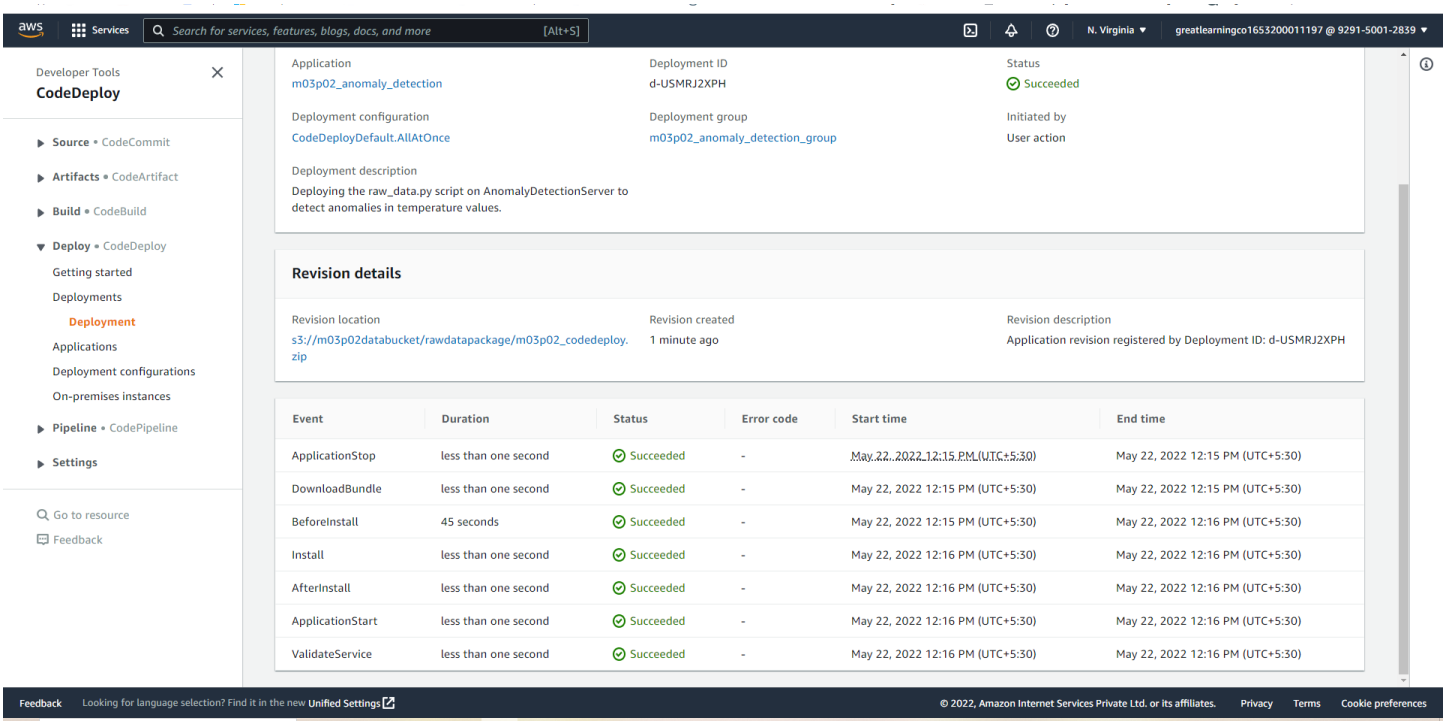
<Screenshot of CodeDeploy application Group creation for TASK 2 >



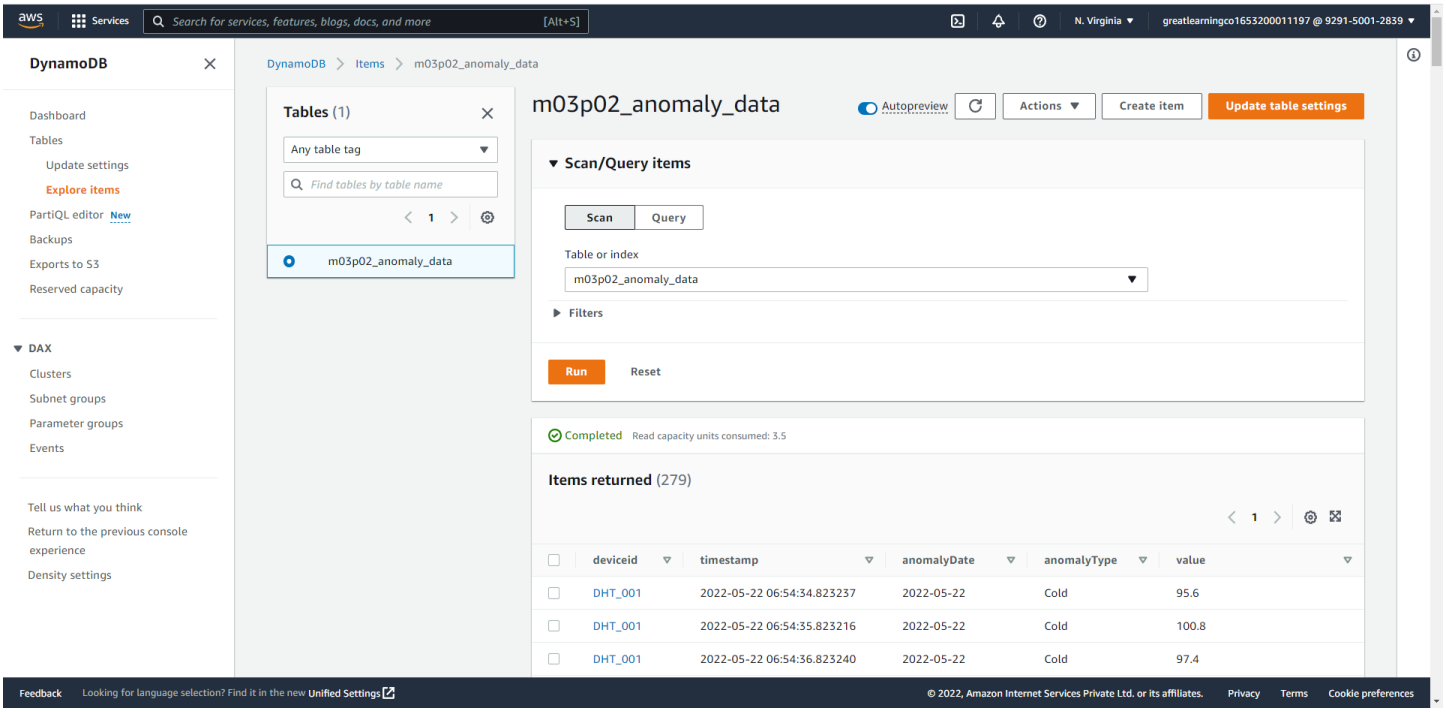
<Screenshot of CodeDeploy Successful c (1) for TASK 2 >



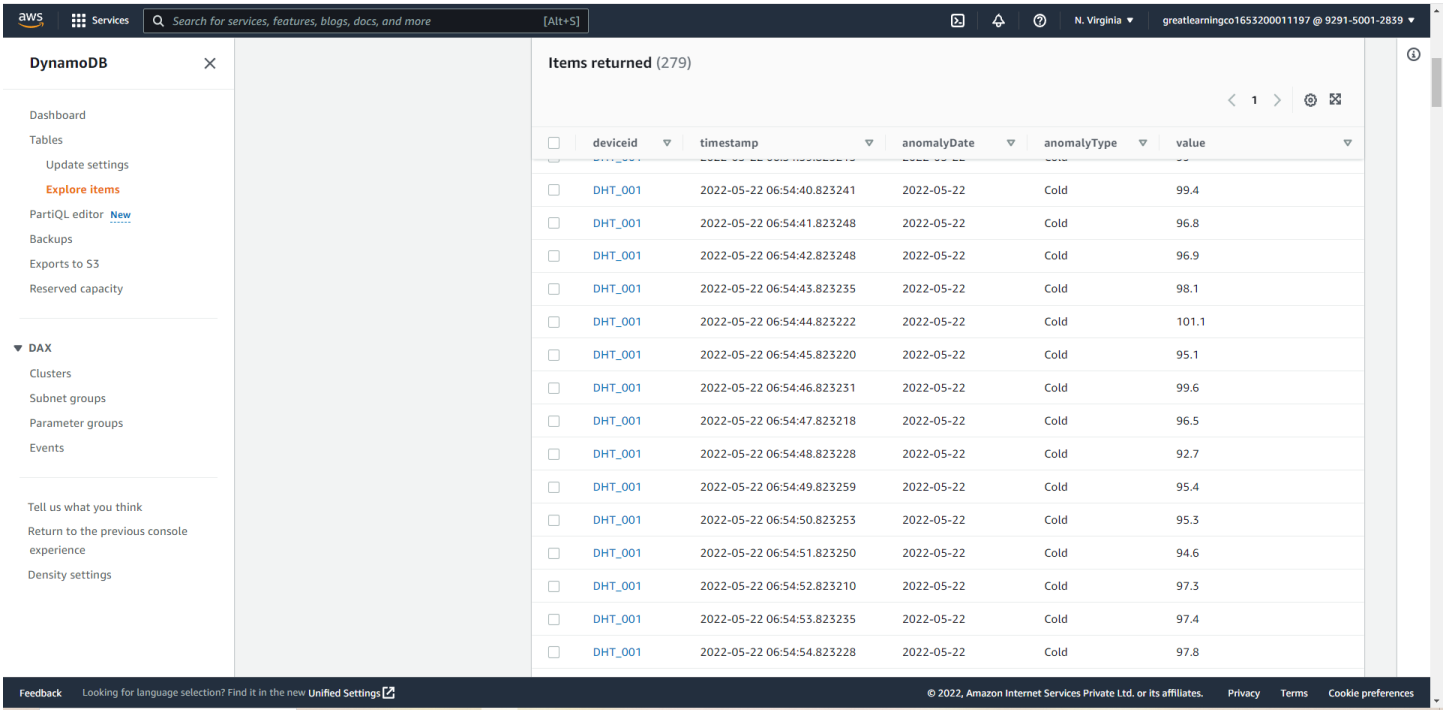
<Screenshot of CodeDeploy Successful c (1) for TASK 2 >



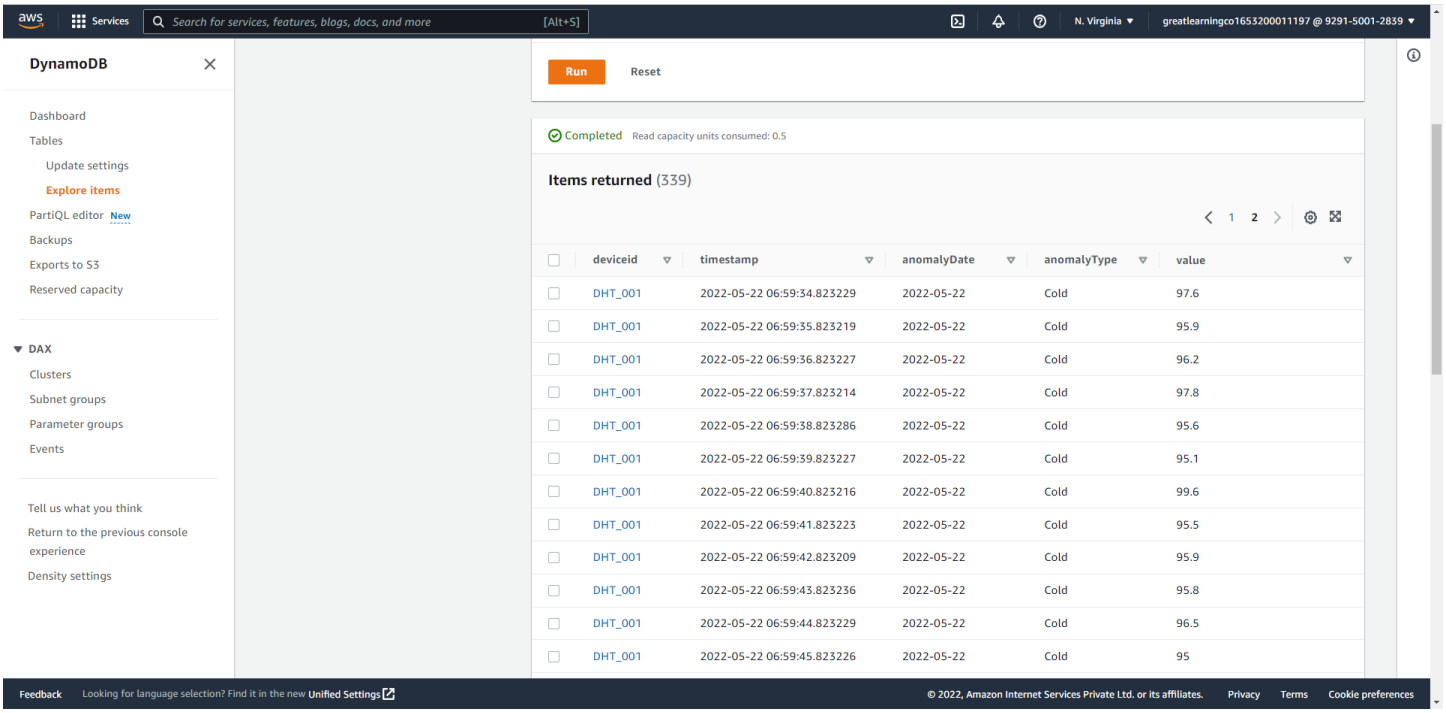
<Screenshot c (2) anomaly data in Dynamodb table for TASK 2 >



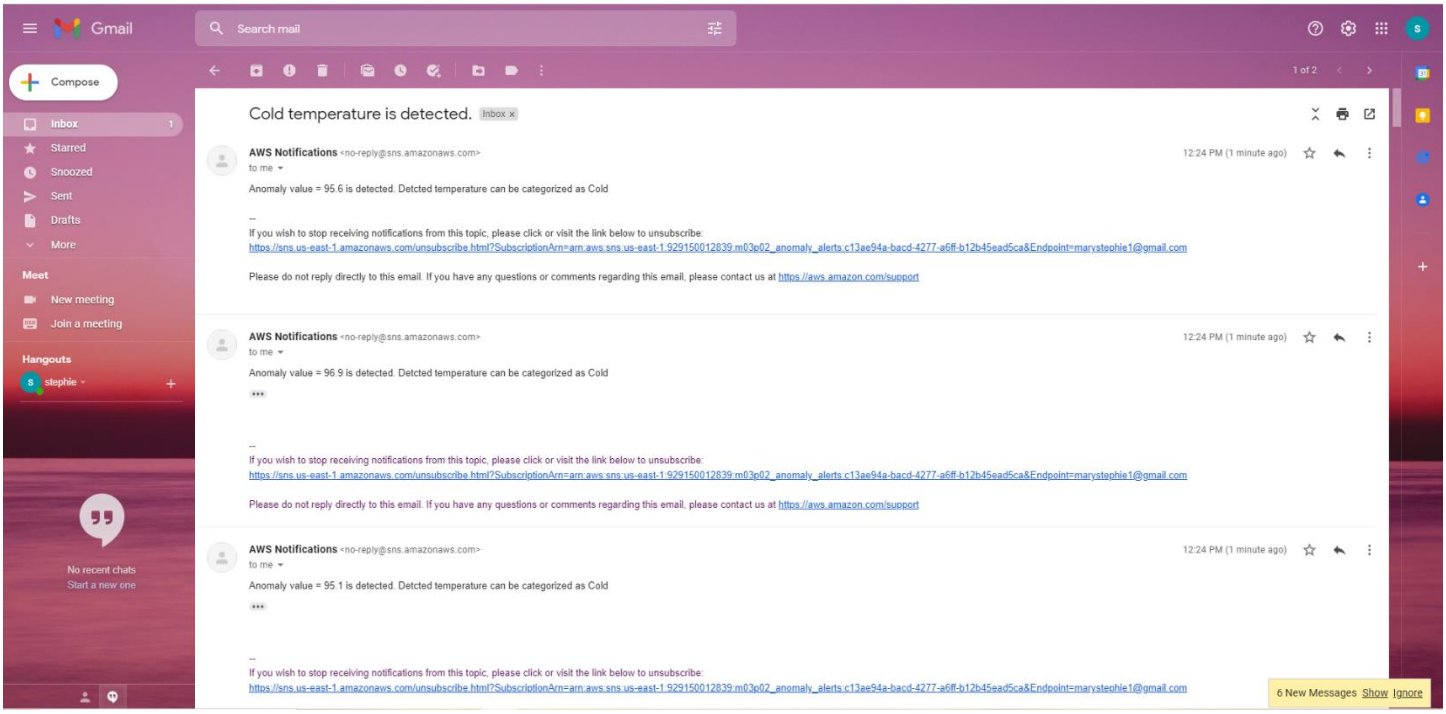
<Screenshot c (2) anomaly data in Dynamodb table for TASK 2 >



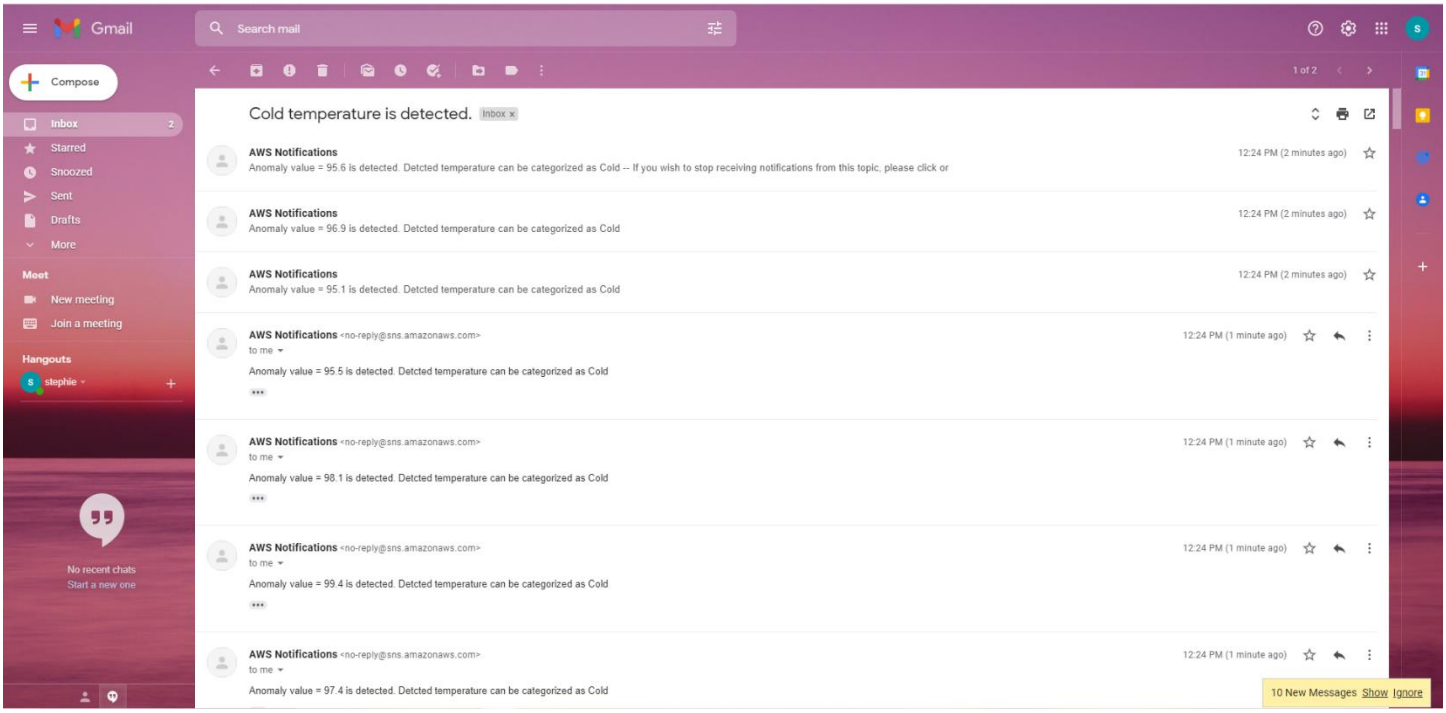
<Screenshot c (2) anomaly data in Dynamodb table for TASK 2 >



<Screenshot c (3) SNS Notification for TASK 2 >



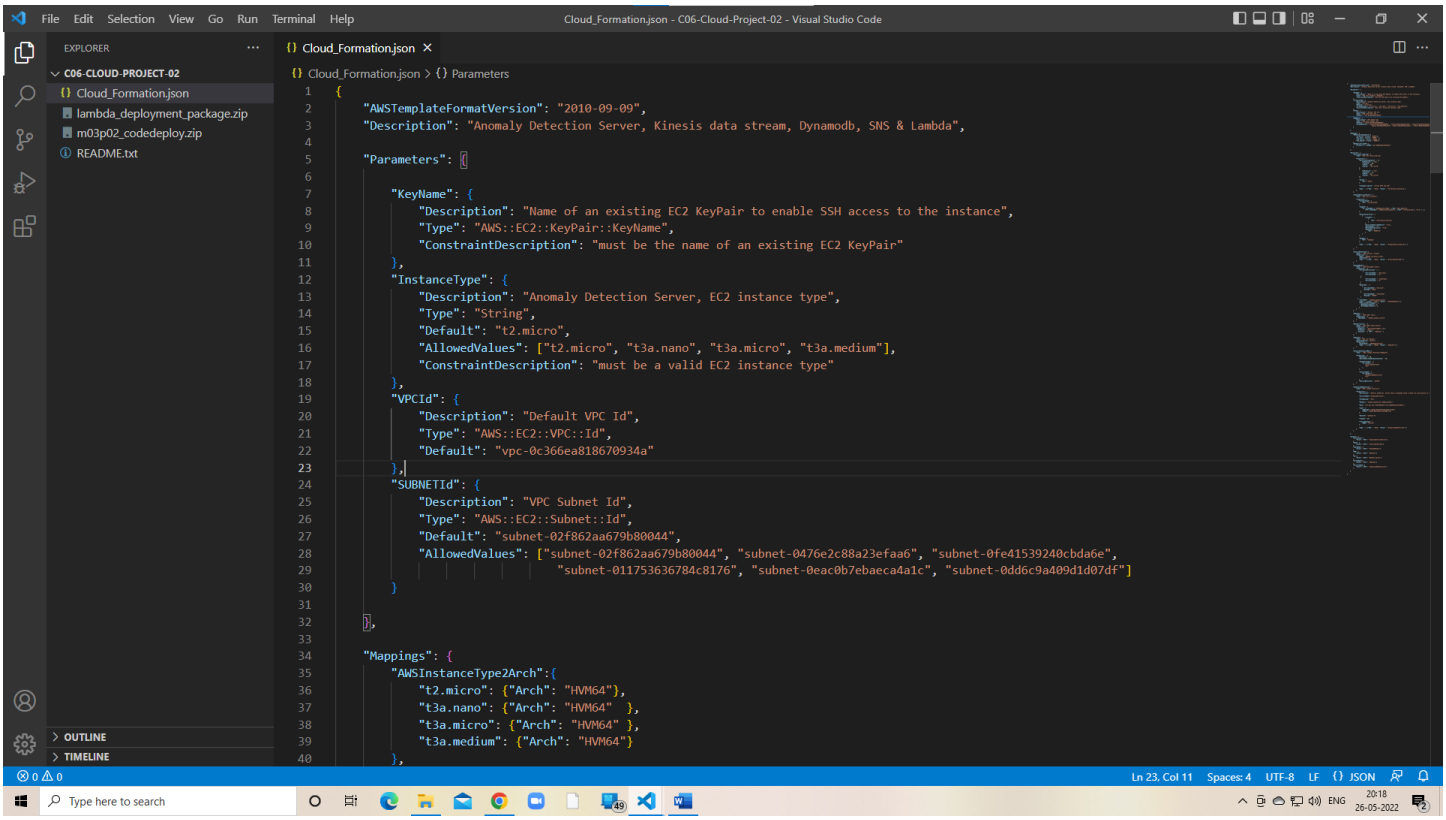
<Screenshot c (3) SNS Notification for TASK 2 >

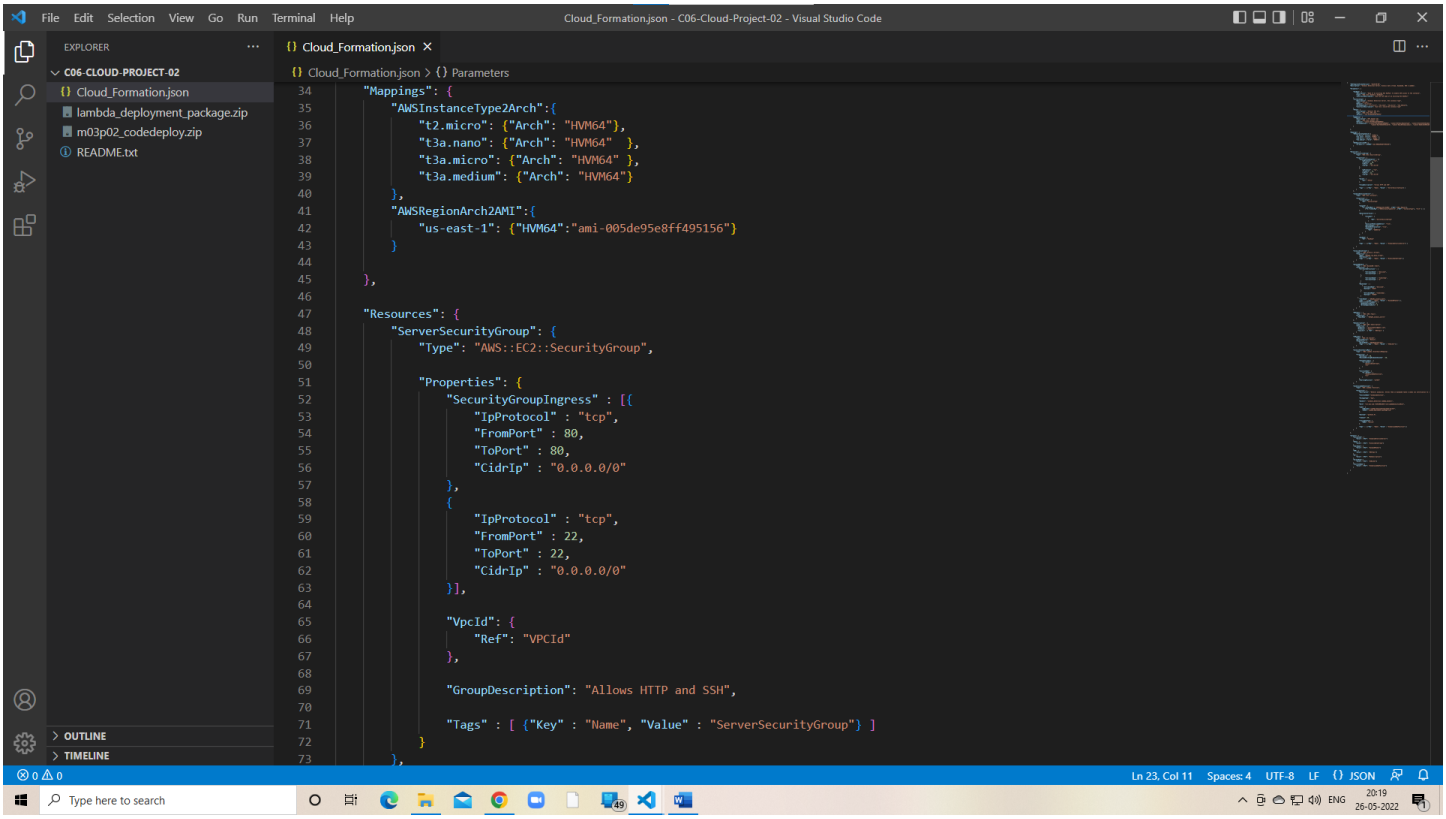


Task 3 Screenshots:

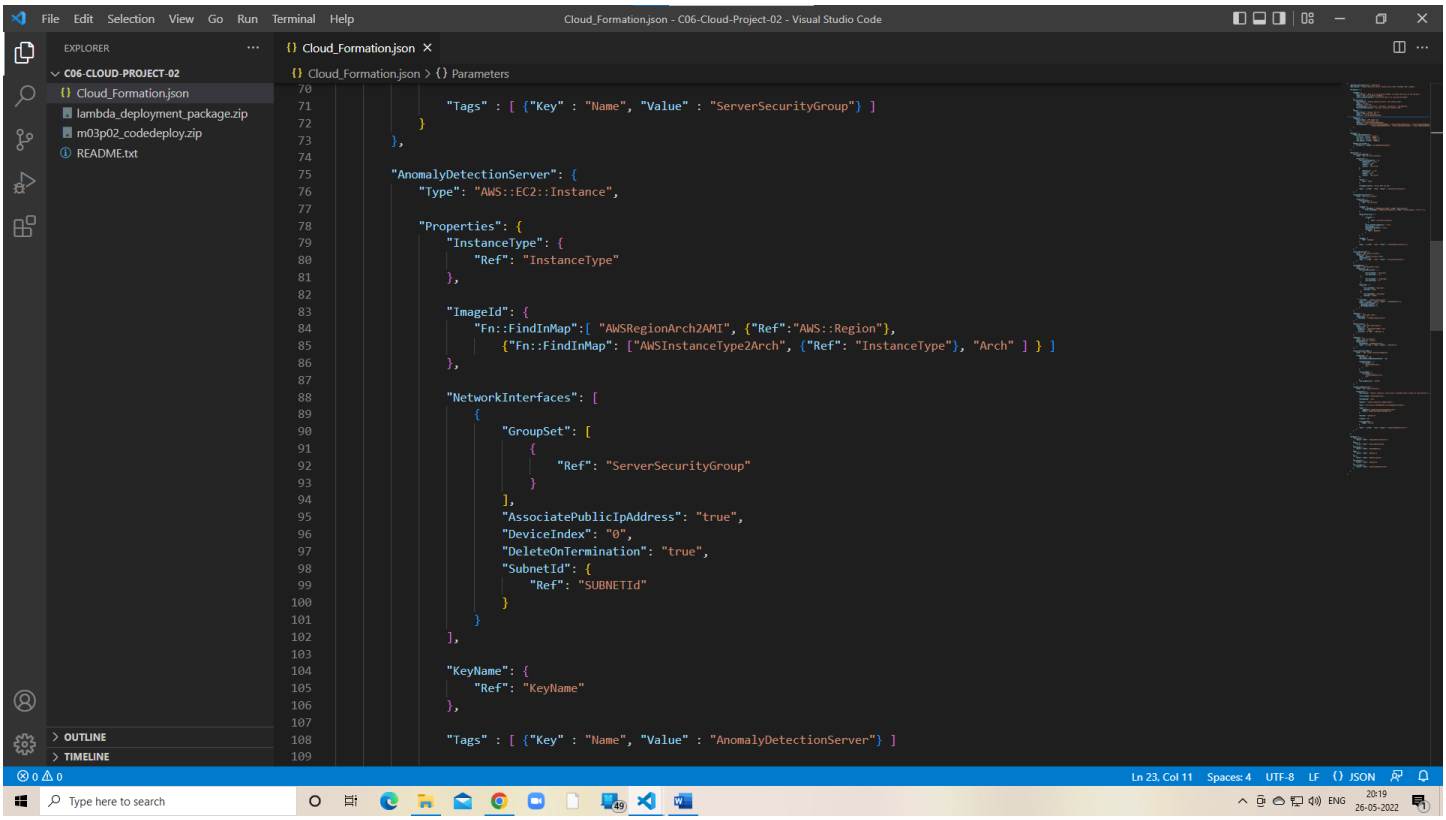
Step number	a
Step name	Cloud formation summary page
Instructions	1) Template file (JSON script) to enable mentioned services in all three tasks
Expected screenshots	1) Cloud formation summary page

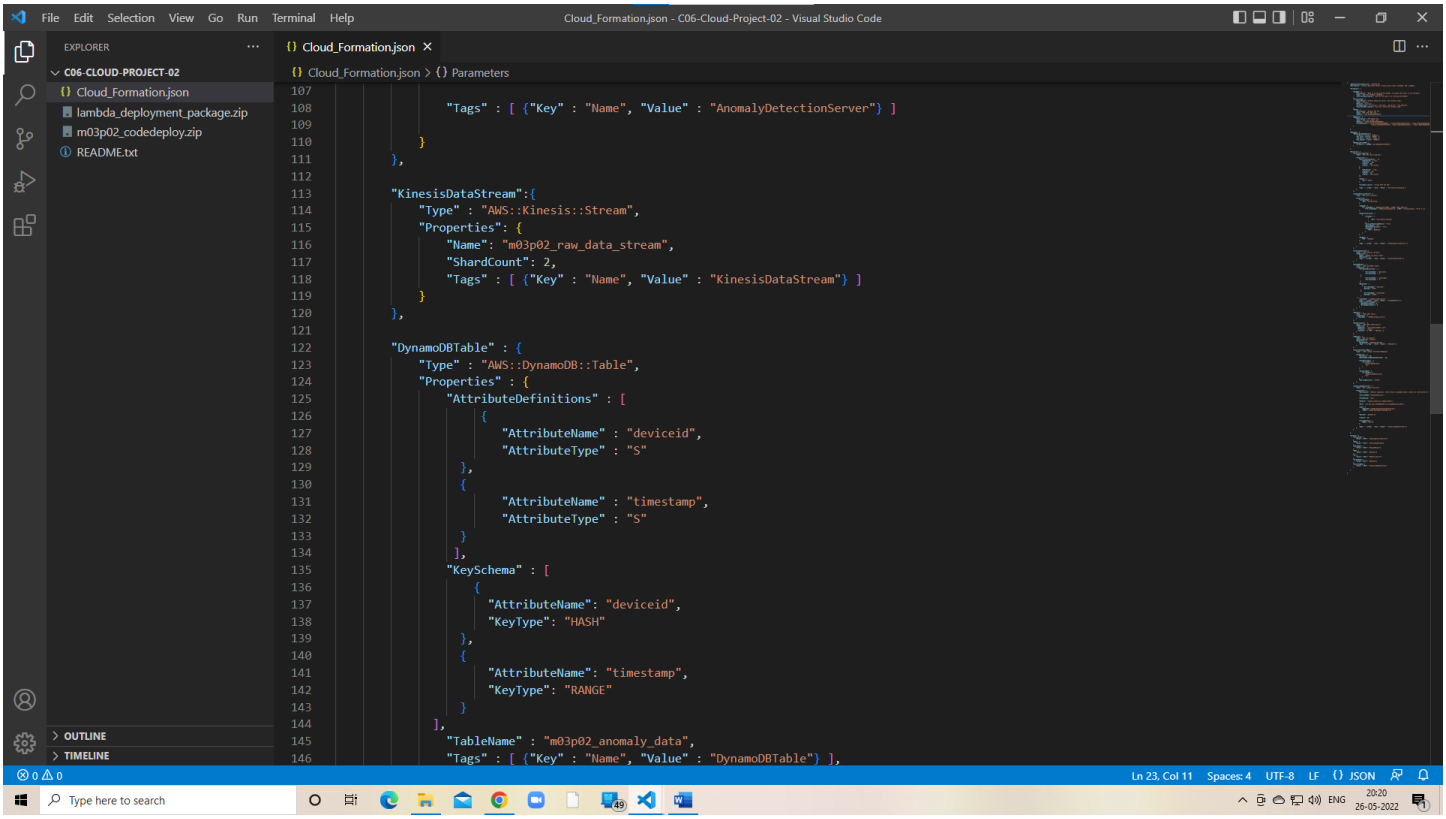
<Screenshot of Cloud_Formation.json template creation for TASK 3>



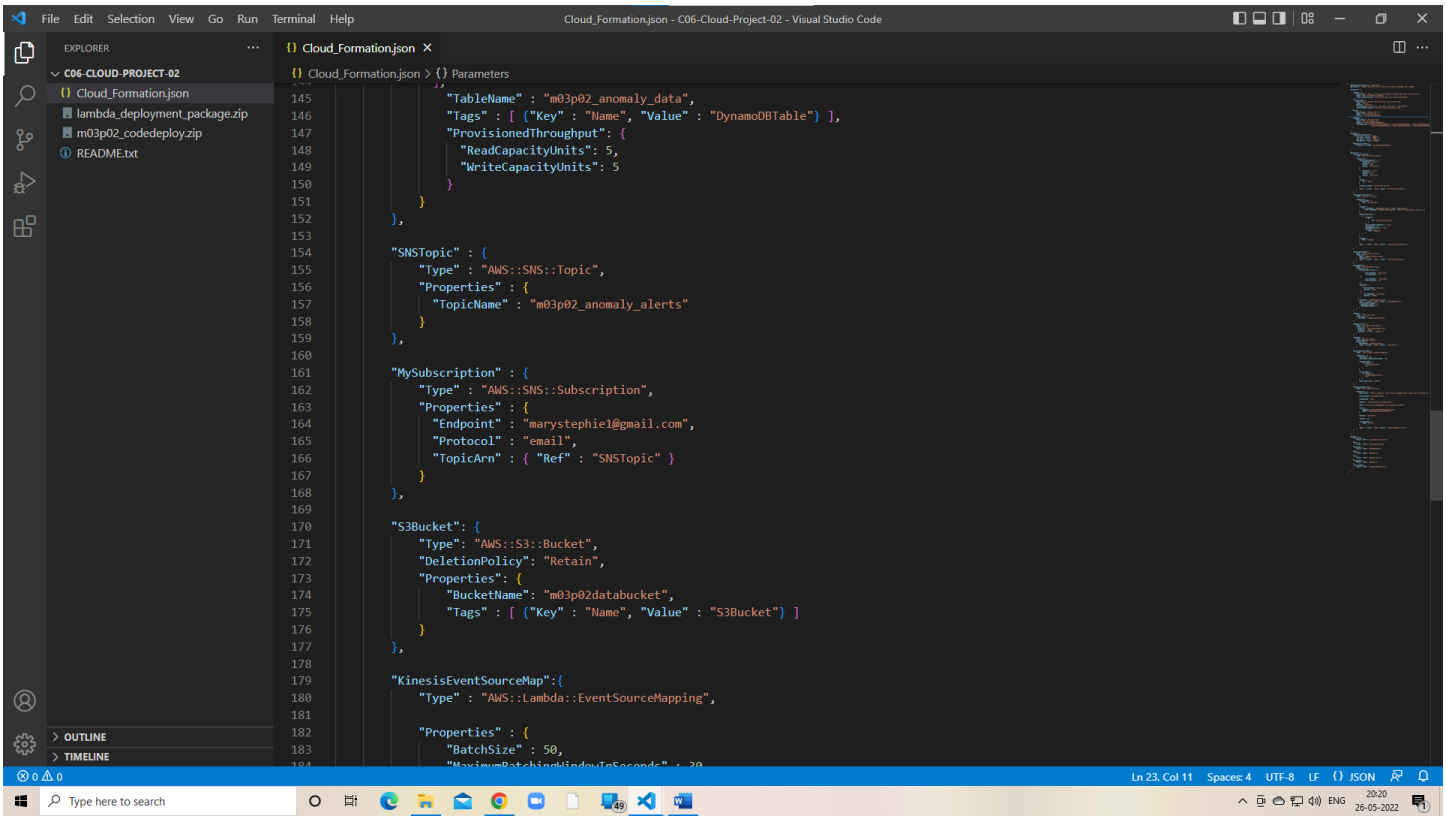


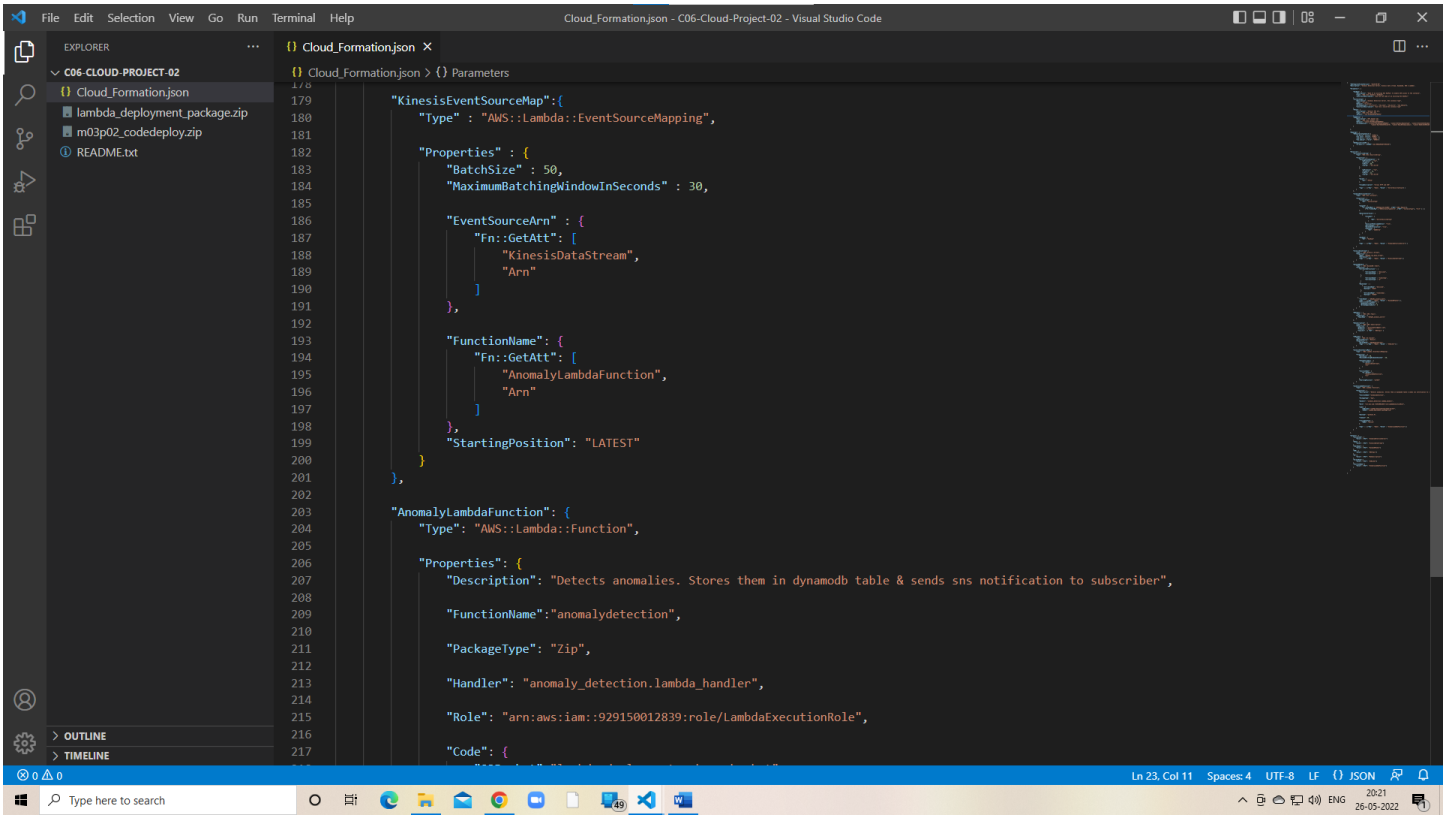
<Screenshot of Cloud_Formation.json template creation for TASK 3>



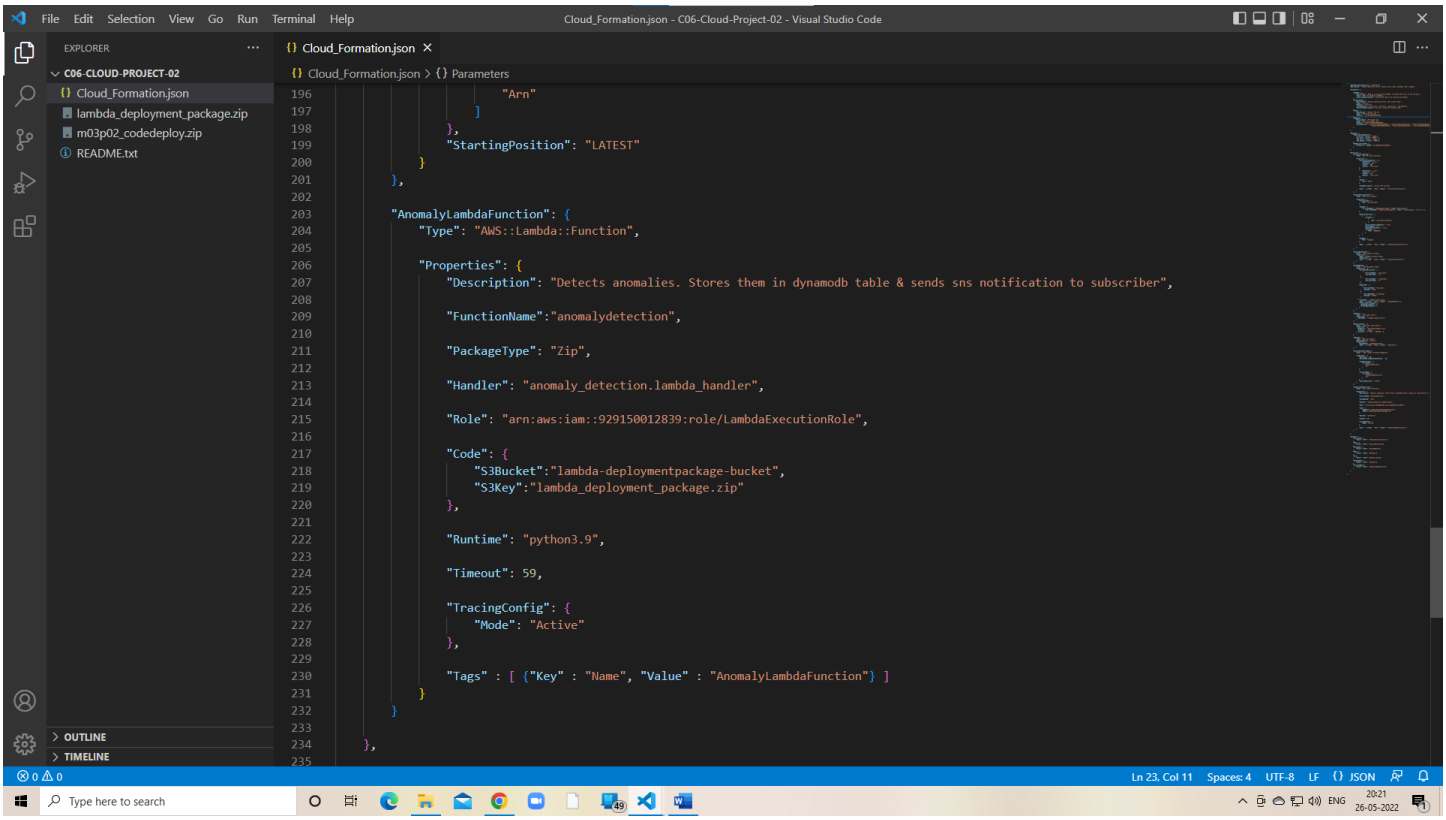


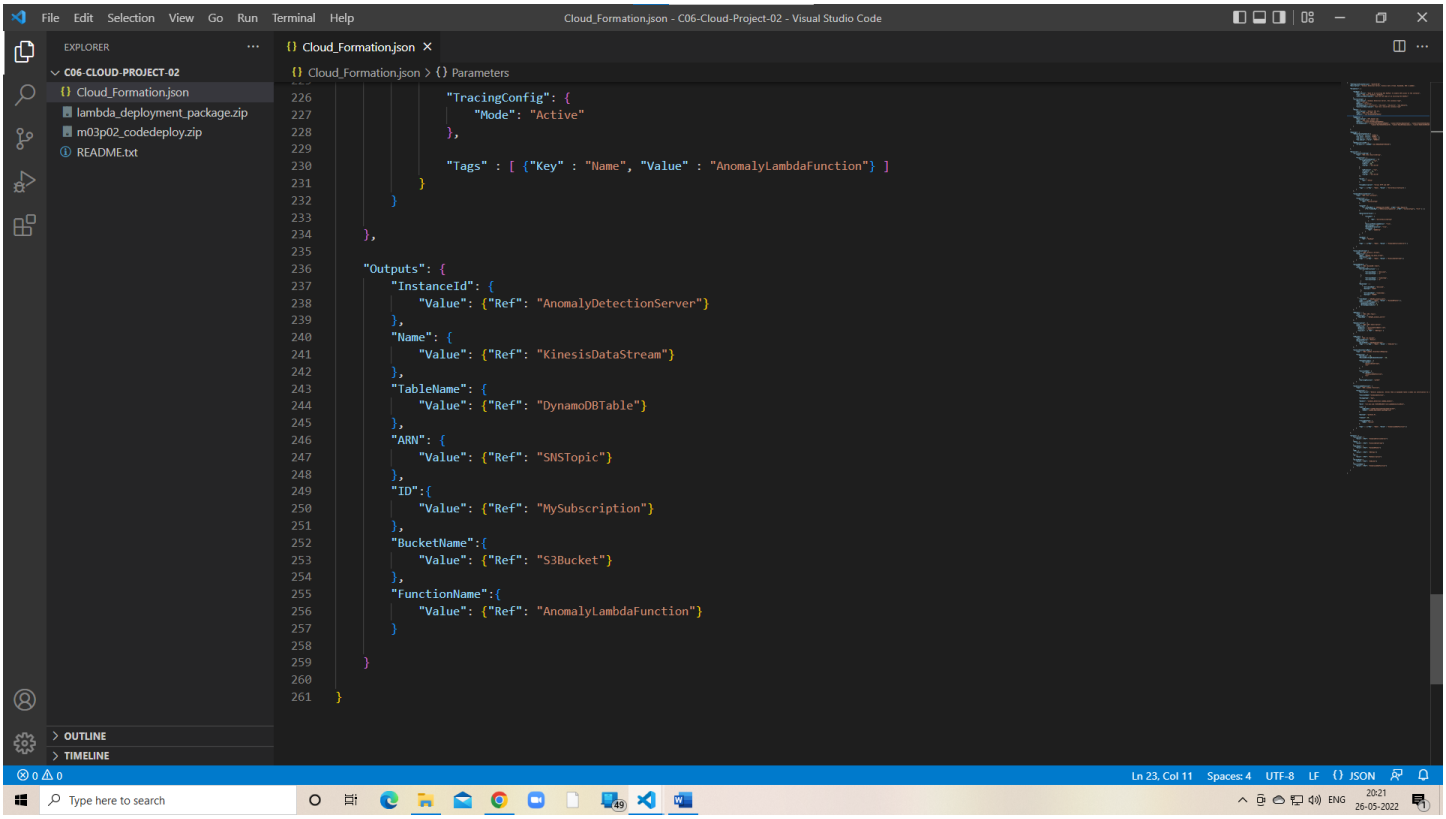
<Screenshot of Cloud_Formation.json template creation for TASK 3>



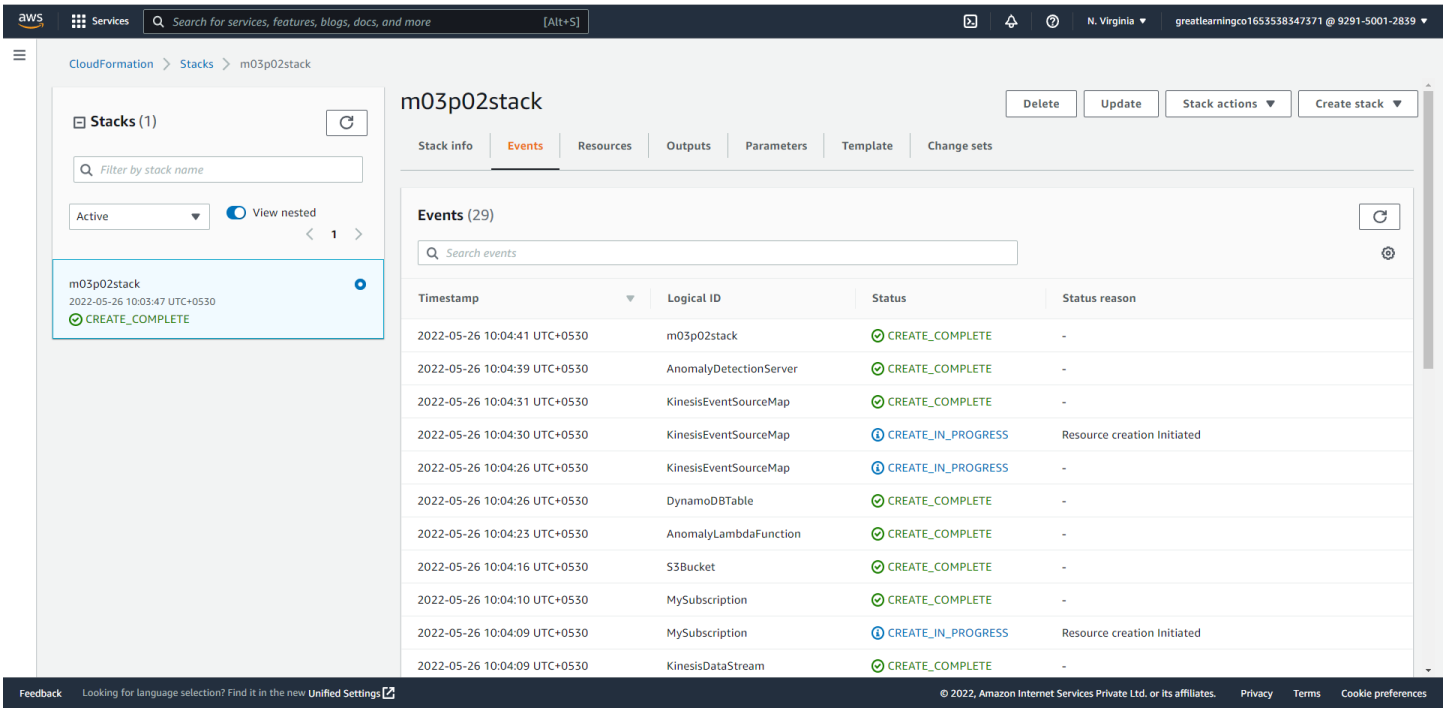


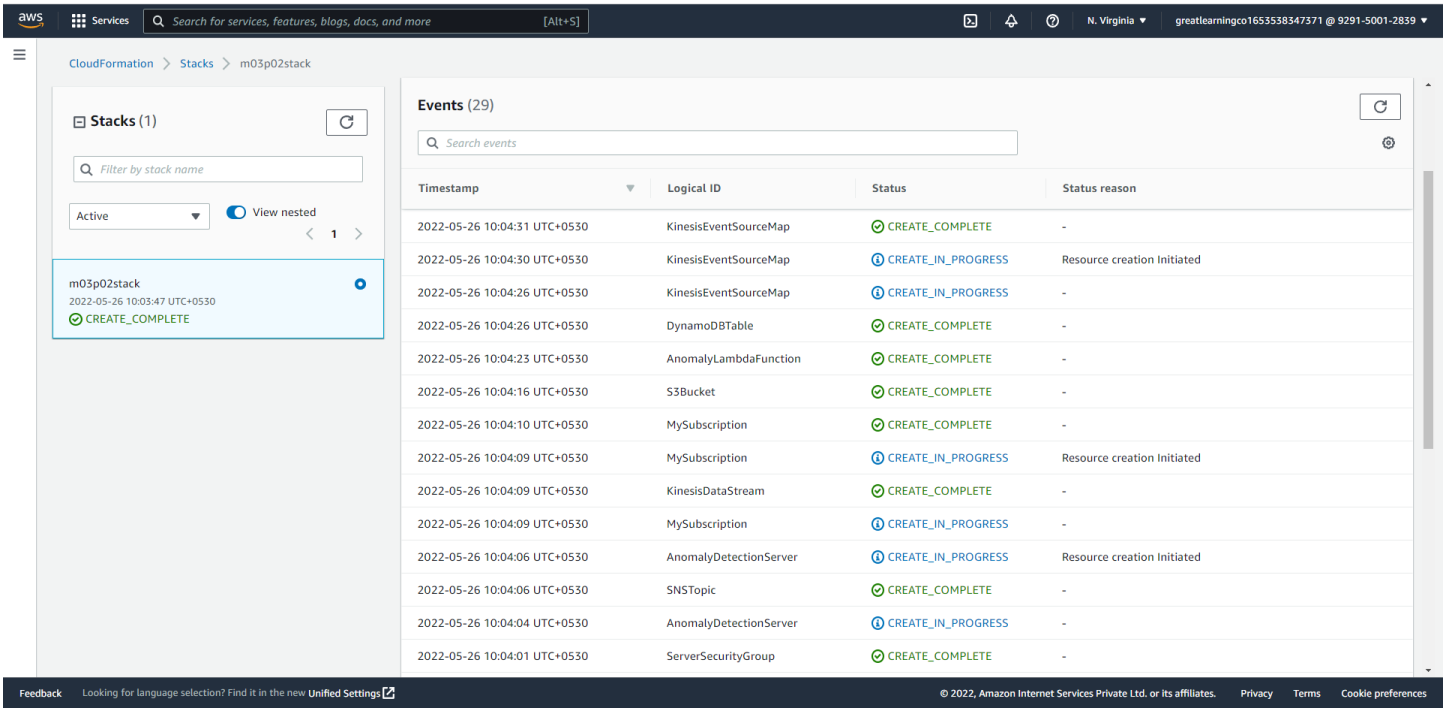
<Screenshot of Cloud_Formation.json template creation for TASK 3>



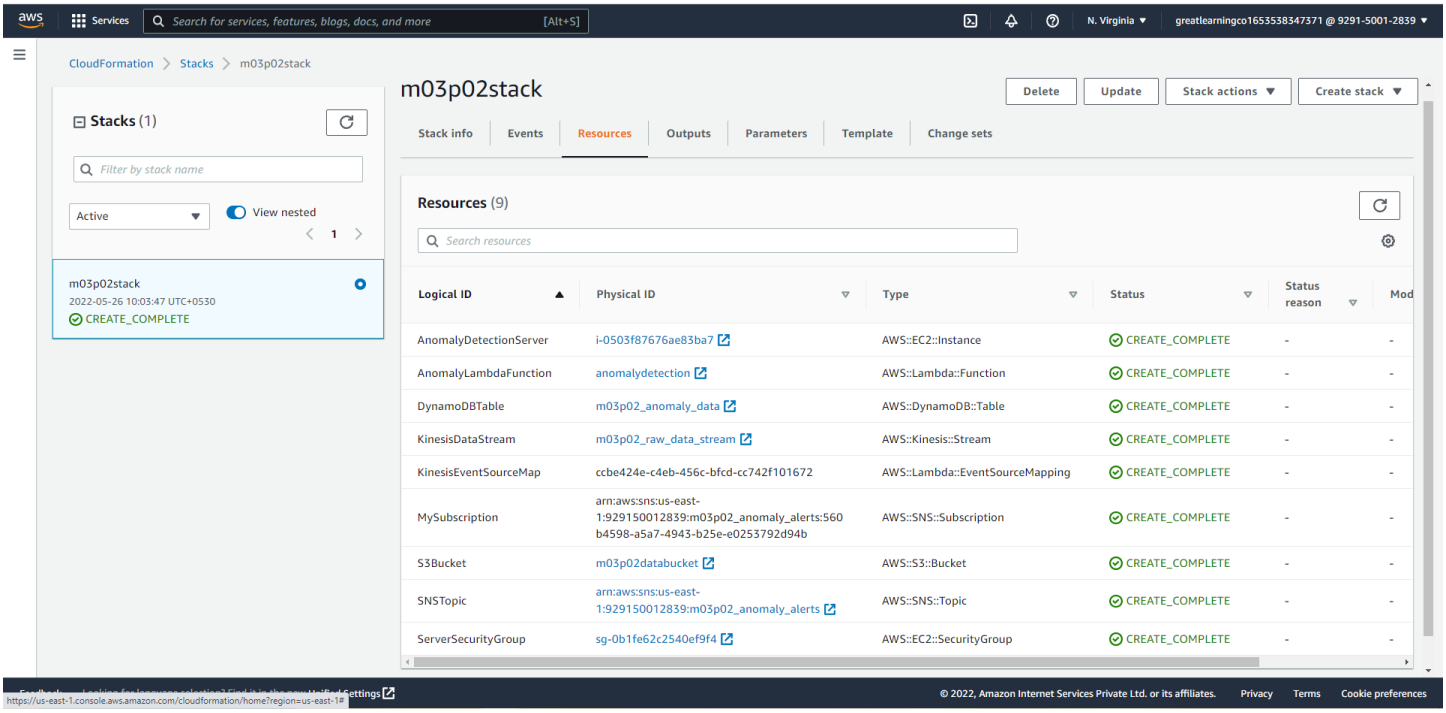


<Screenshot a (1) Cloud “m03p02stack” successful creation for TASK 3>

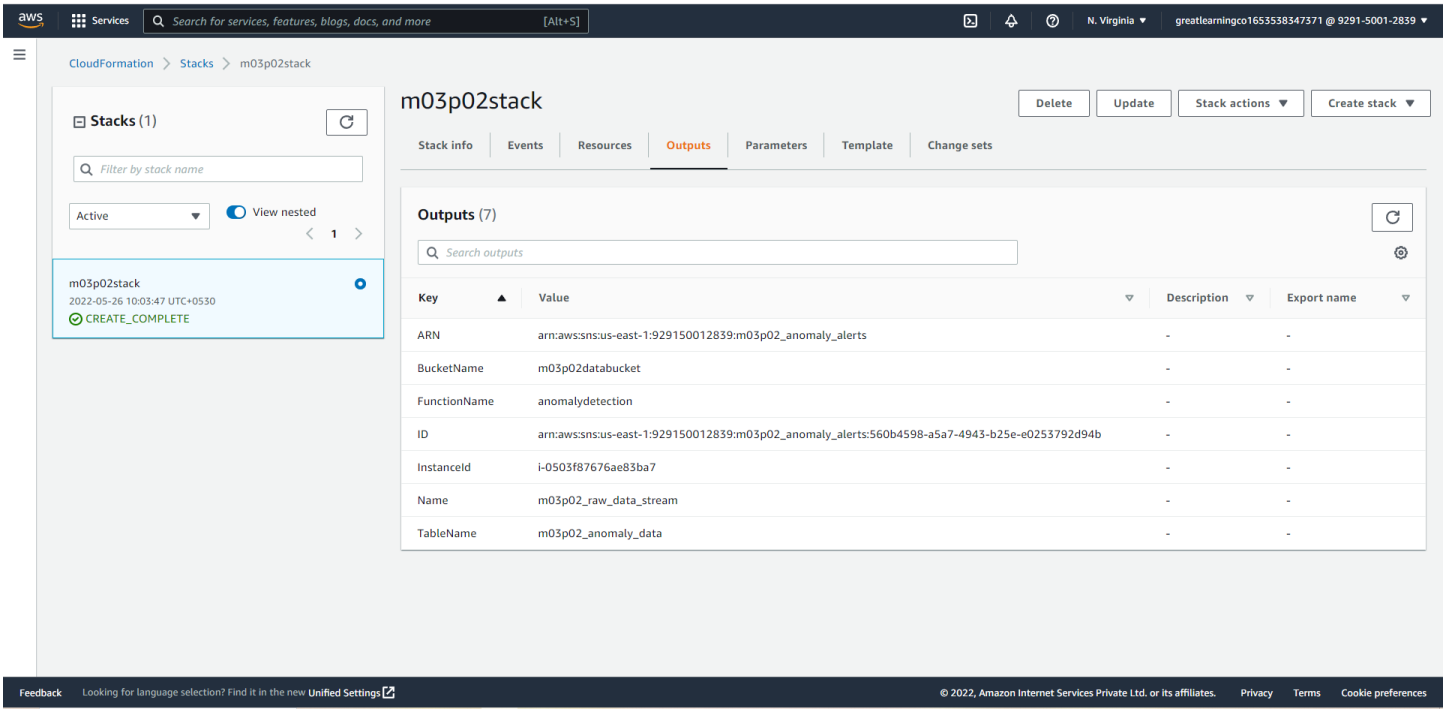




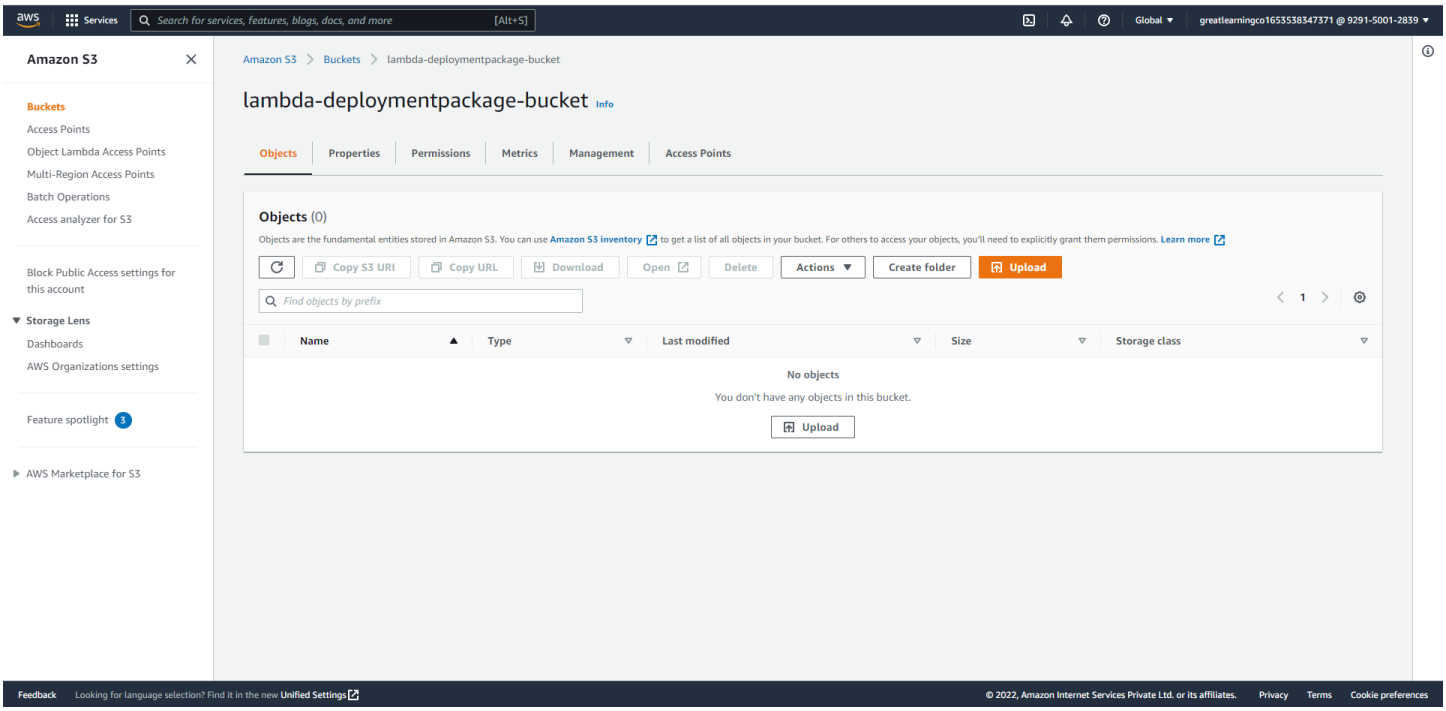
<Screenshot a (1) Cloud “m03p02stack” successful creation of “Resources” for TASK 3>



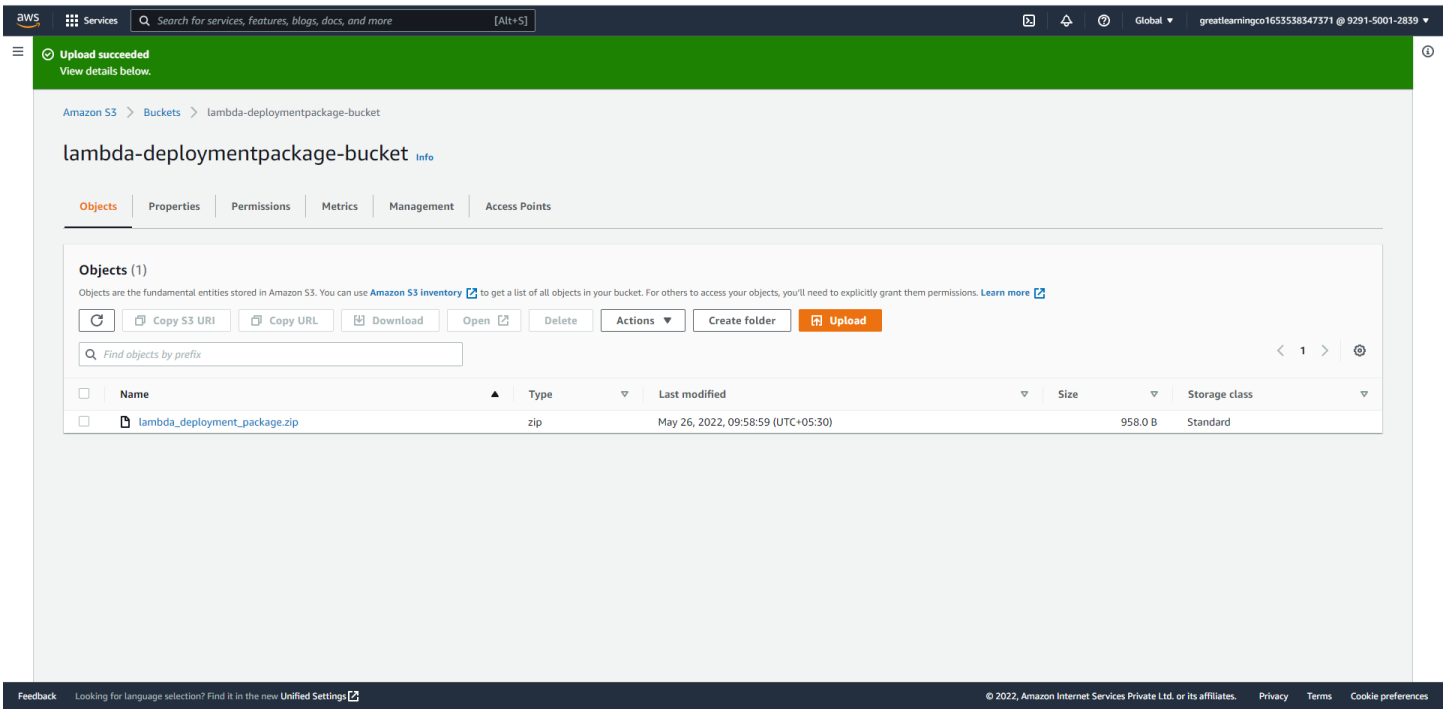
<Screenshot a (1) Cloud “m03p02stack” successful “Outputs” for TASK 3>



<Screenshot of separate S3 “lambda-deploymentpackage-bucket” created to upload lambda zip file for TASK 3>

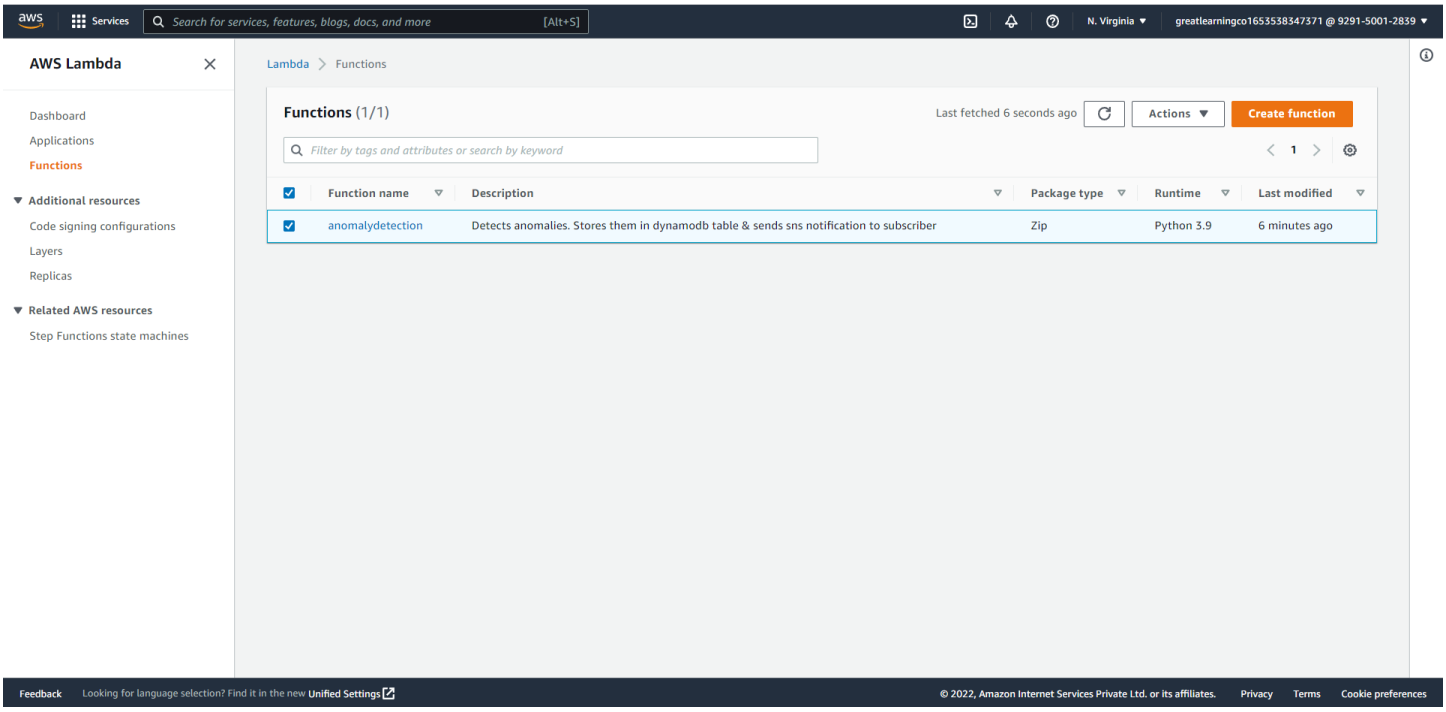


<Screenshot of uploading “lambda_deployment_package.zip” to lambda-deploymentpackage-bucket for TASK 3 >

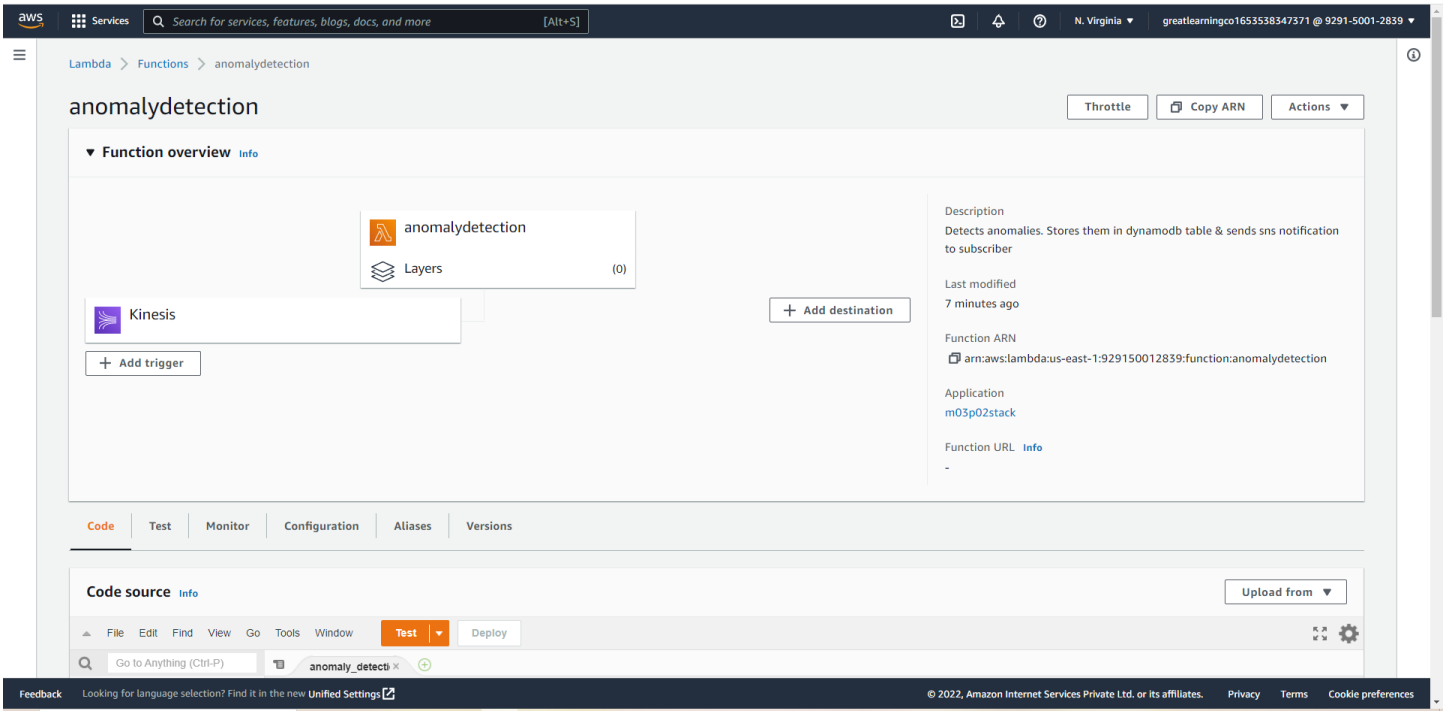


Step number	b
Step name	Enabled lambda handler service
Instructions	1) Make sure that cloud formation operation is successful 2) Check if the lambda handler service is successfully enabled
Expected screenshots	1) Lambda handler

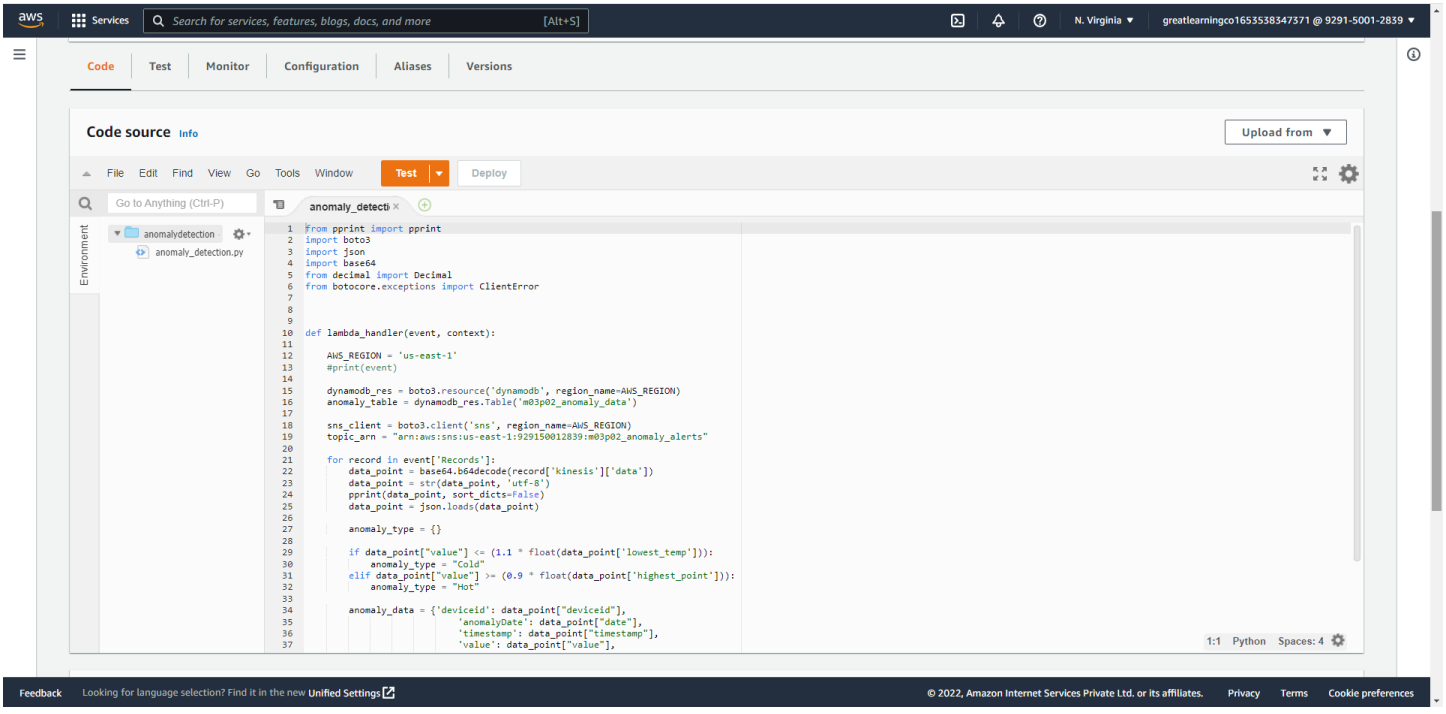
<Screenshot b (1) Lambda Handler successfully created through Cloudformation for TASK 3>



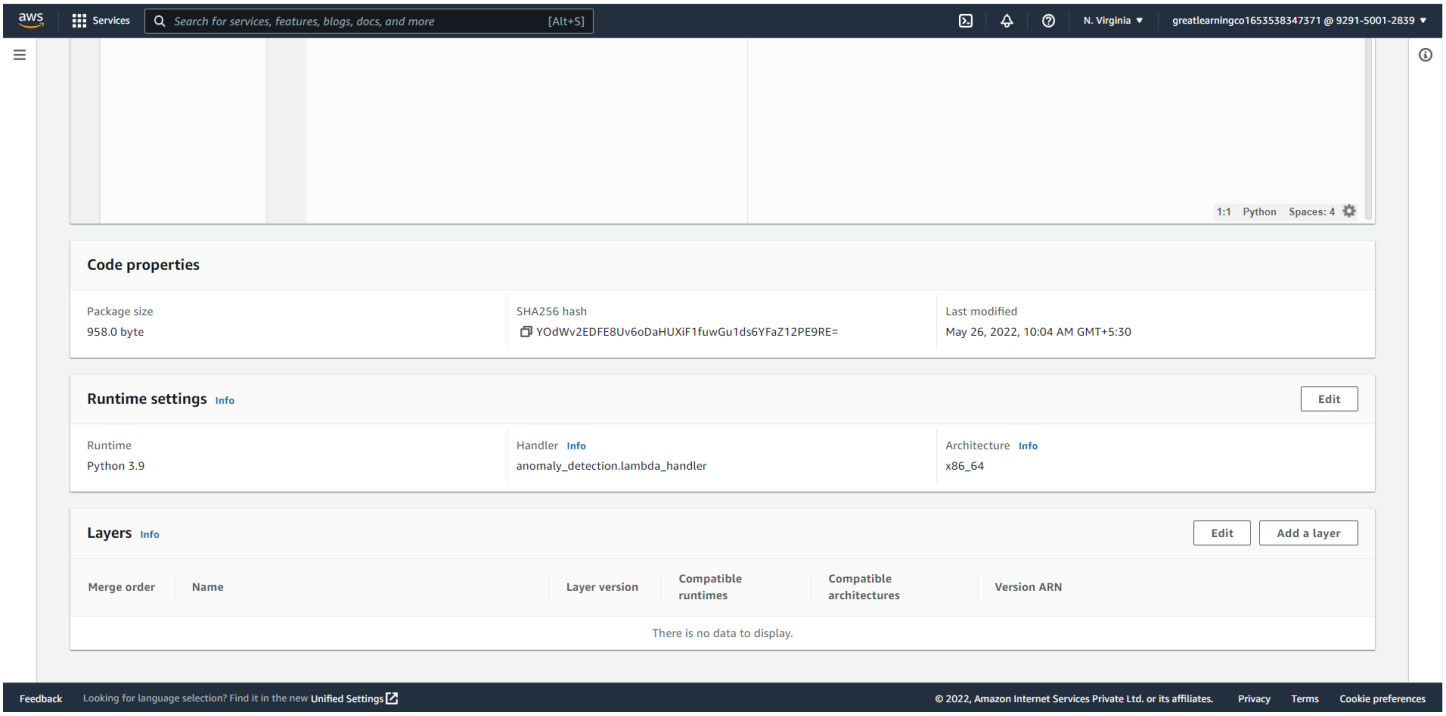
<Screenshot b (1) Lambda Handler successfully created through Cloudformation for TASK 3>



<Screenshot b (1) Lambda Handler successfully created through Cloudformation for TASK 3>

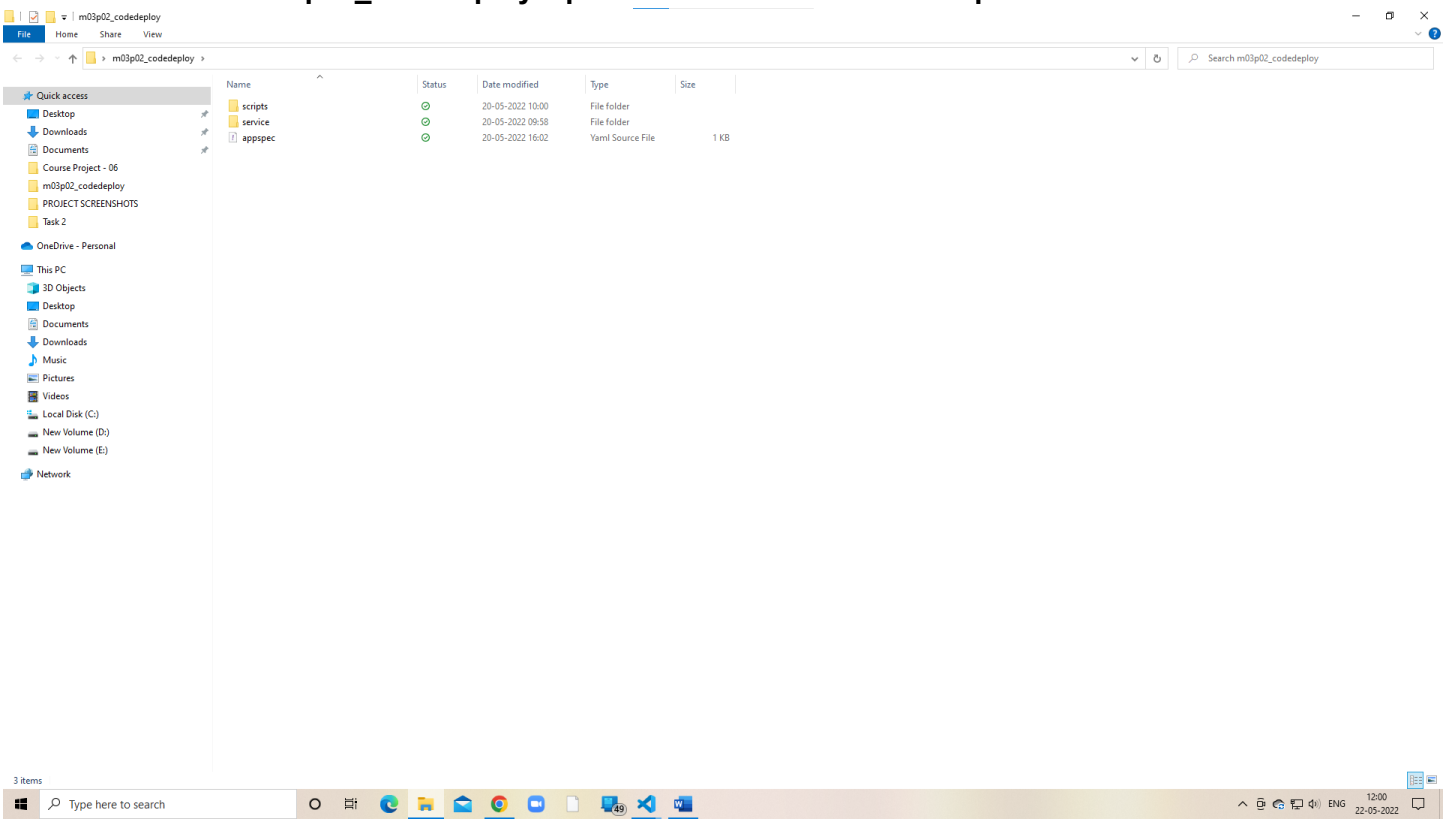


<Screenshot b (1) Lambda Handler successfully created through Cloudformation for TASK 3>

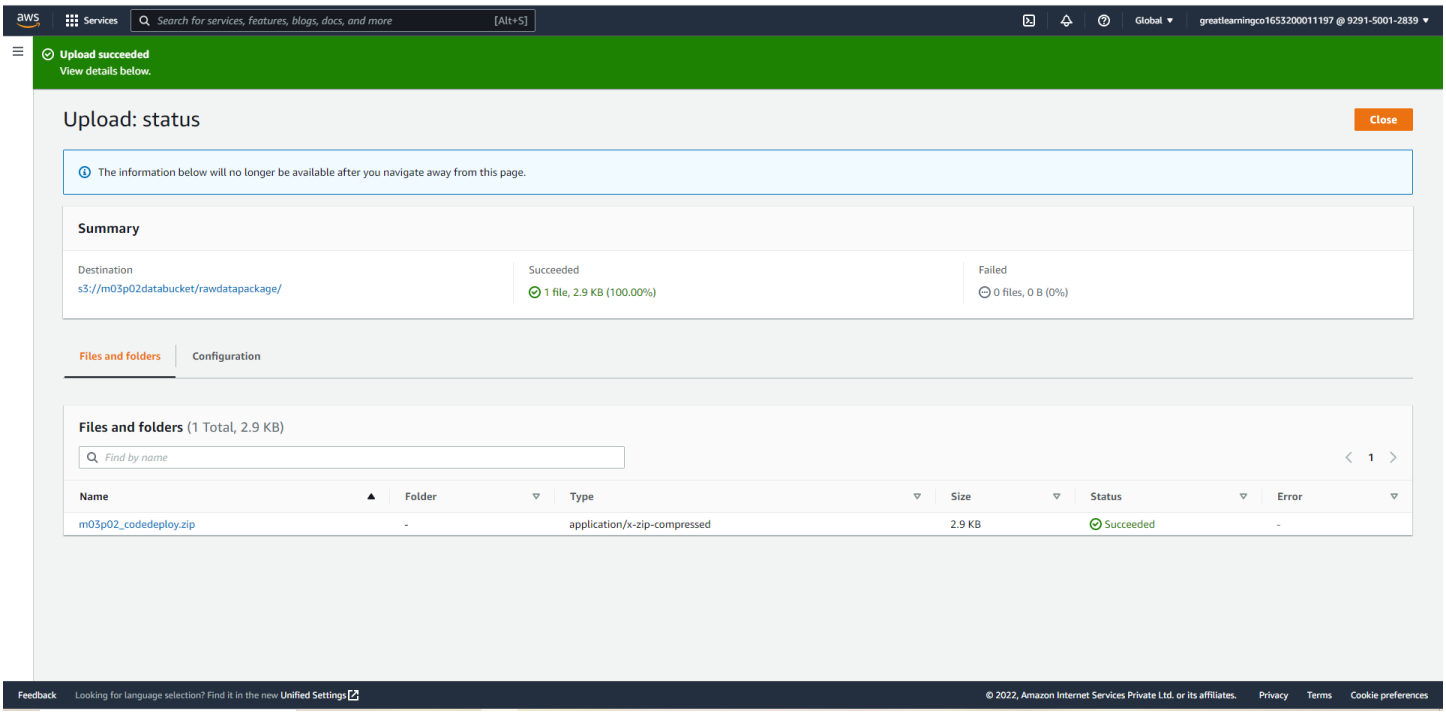


Step number	c
Step name	CodeDeploy
Instructions	3) Perform the CodeDeploy operation through AWS console 4) If CodeDeploy is successful look if the Anomaly detection related data is going into the relevant DB and SNS services
Expected screenshots	4) Successful CodeDeploy operation summary page 5) Anomaly data pushed in the DynamoDB 6) SNS notification

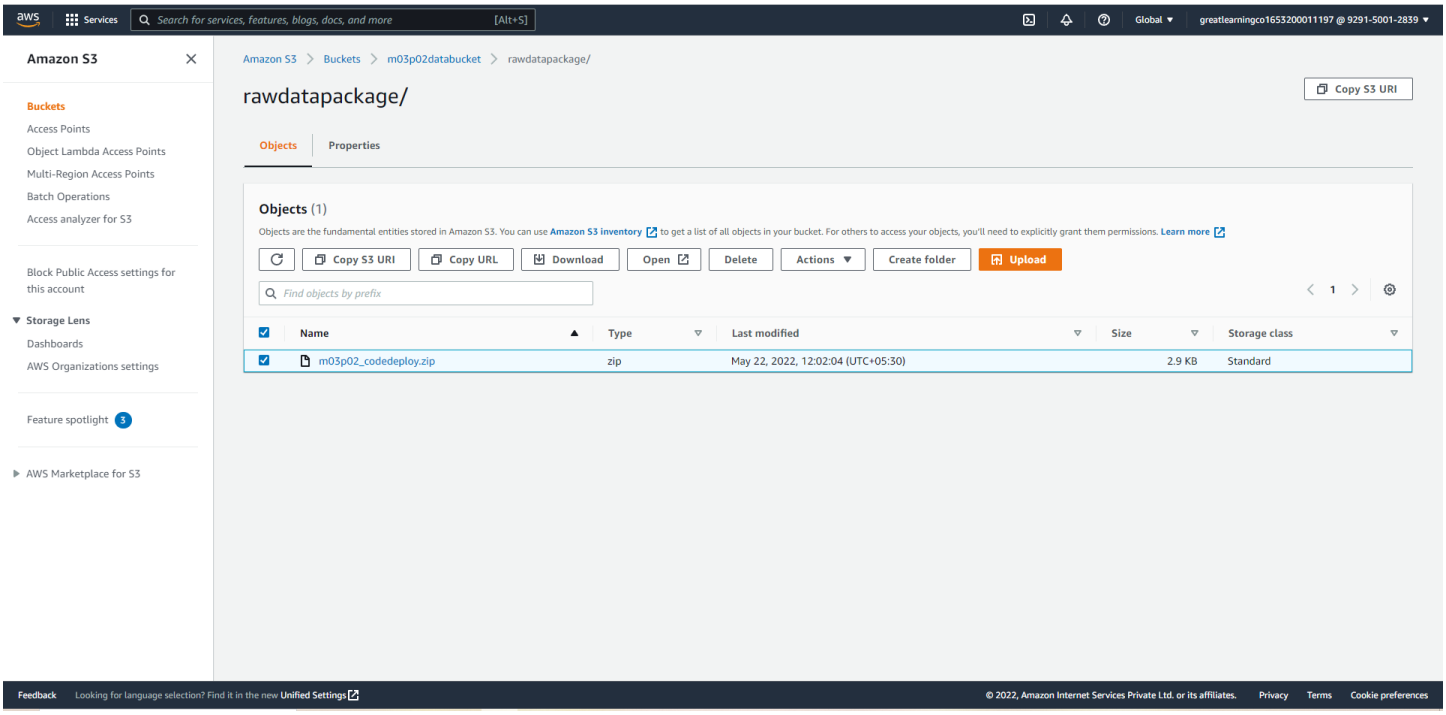
< Screenshot of “m03p02_codedeploy.zip” folder which needs to be uploaded in S3 bucket for TASK 3>



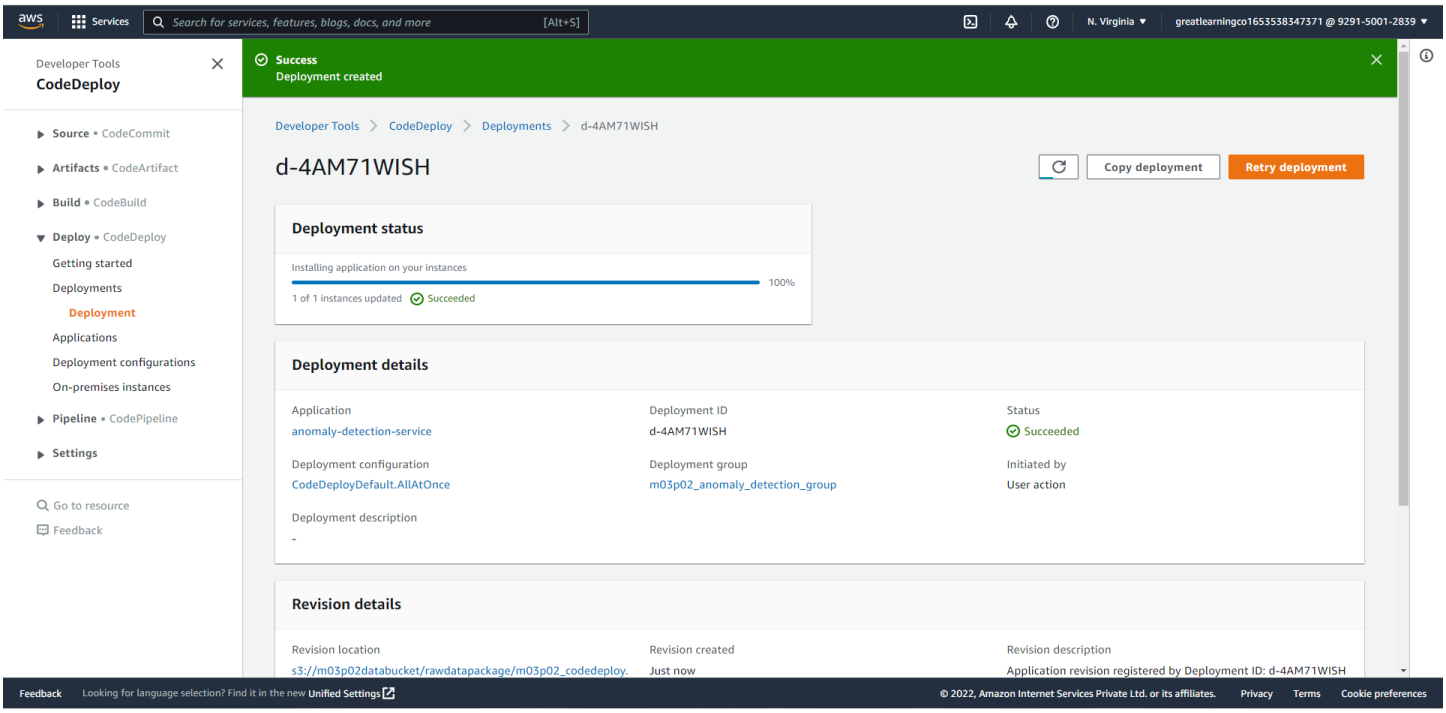
<Screenshot of “m03p02_codedeploy.zip” which contains both application and appspec.yml uploaded to m03p02bucket for TASK 3>



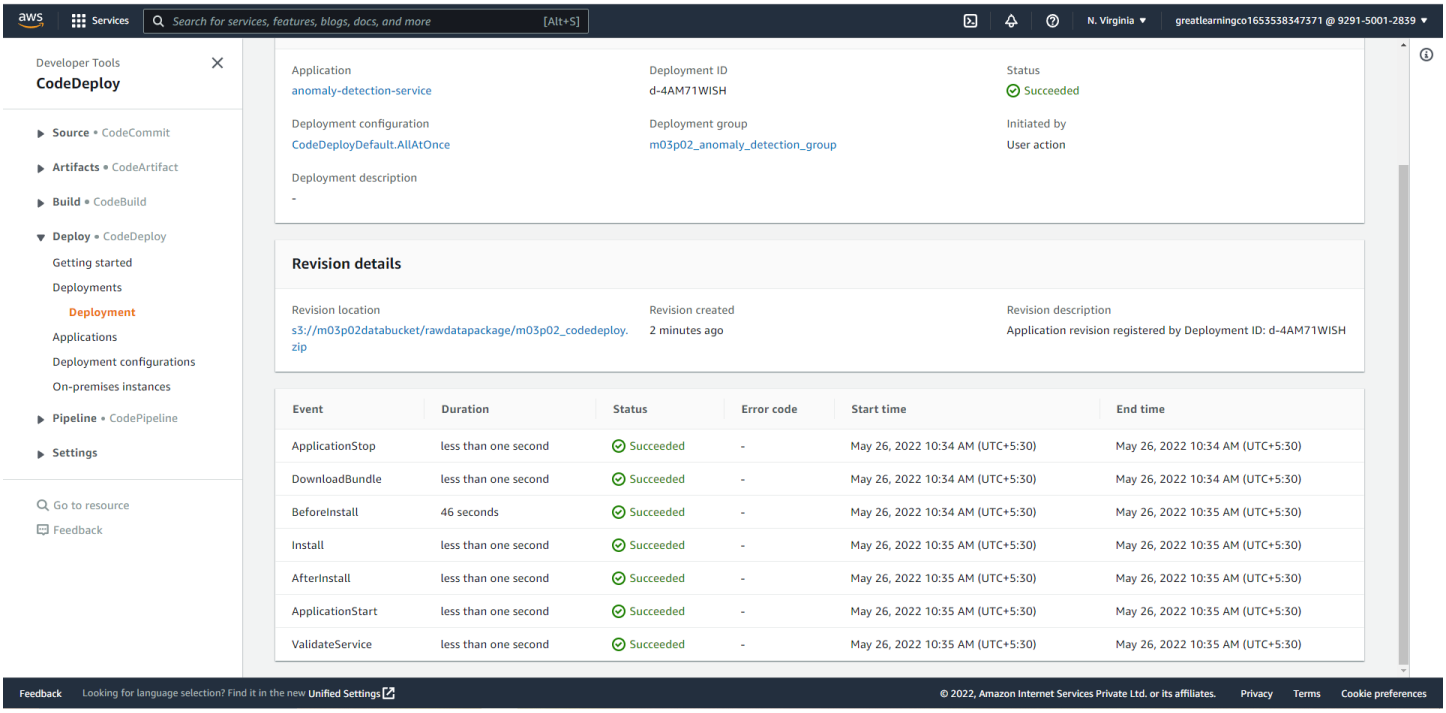
<Screenshot of “m03p02_codedeploy.zip” which contains both application and appspec.yml uploaded for TASK 3>



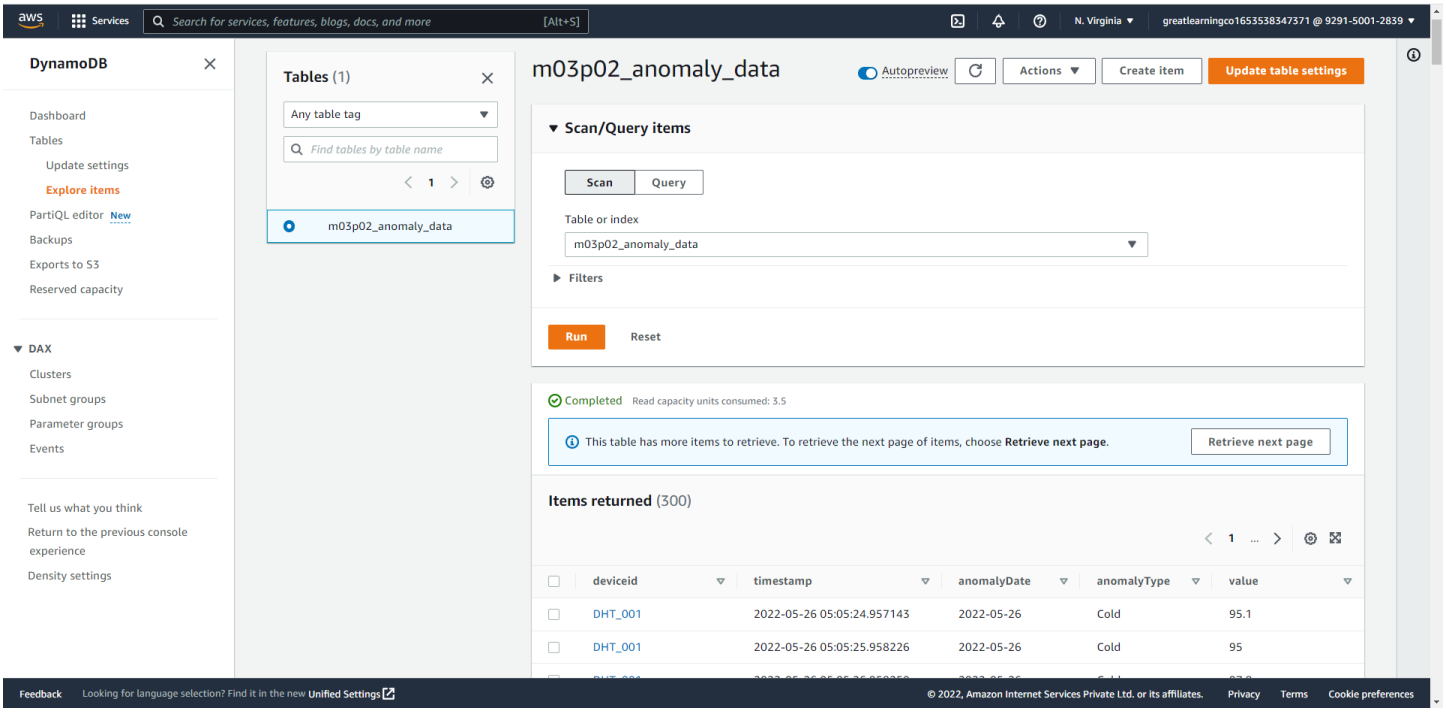
<Screenshot c (4) of Successful CodeDeploy Summary page for TASK 3 >



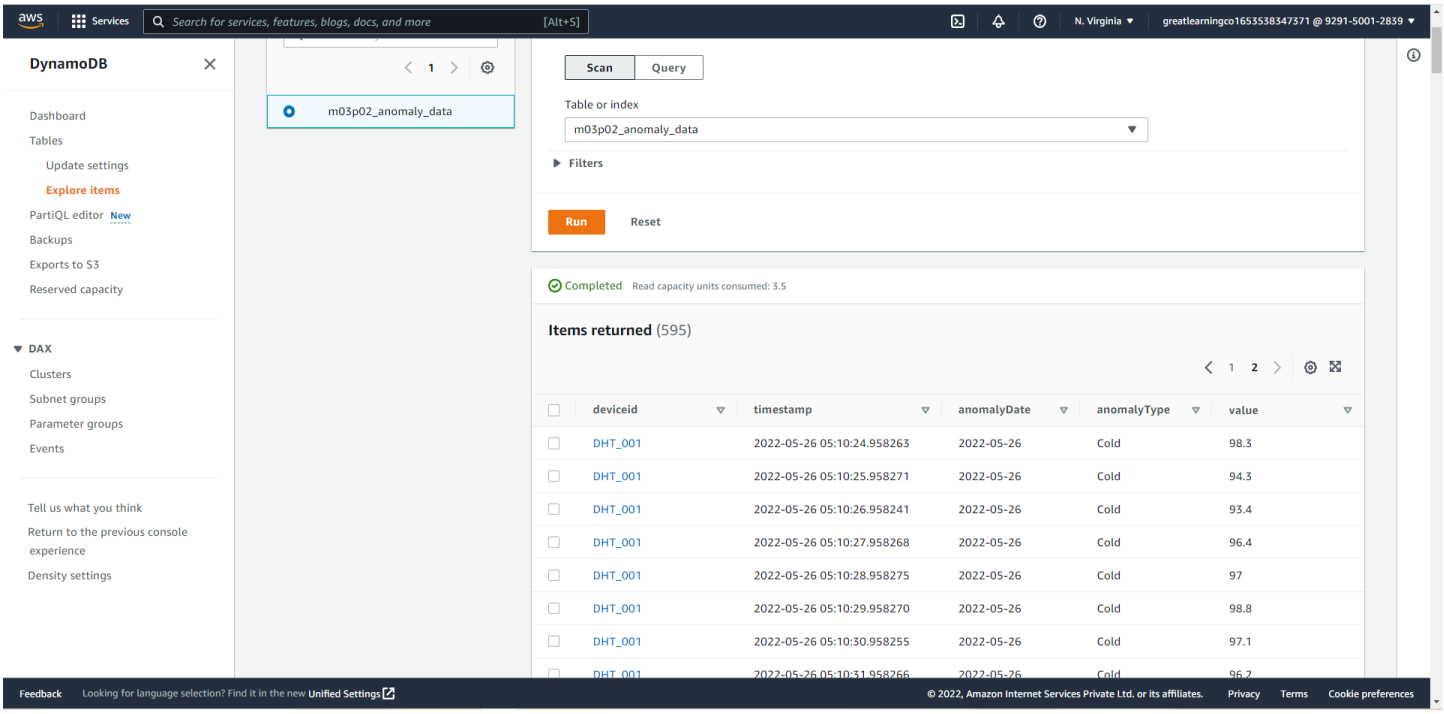
<Screenshot c(4) of Successful CodeDeploy for TASK 3>



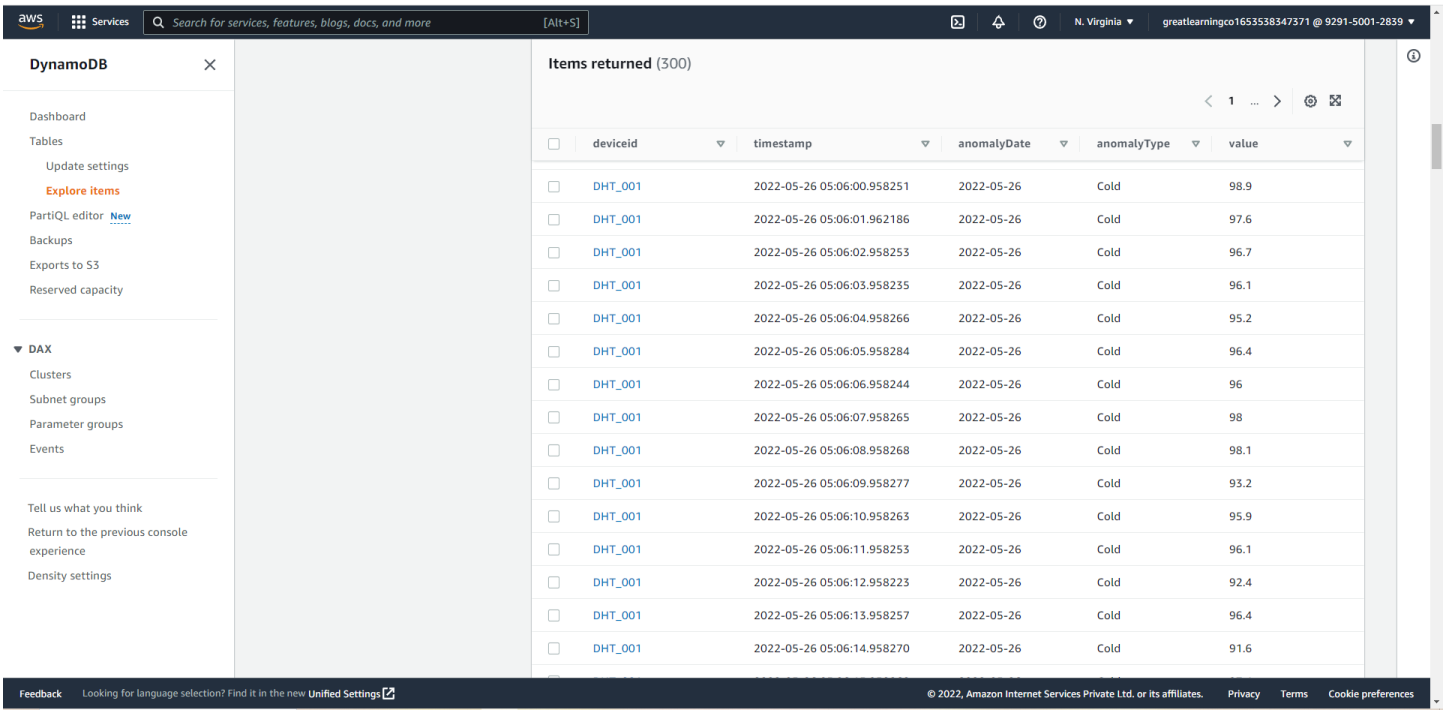
< Screenshot c(5) of Anomaly Data pushed in Dynamodb table for TASK 3>



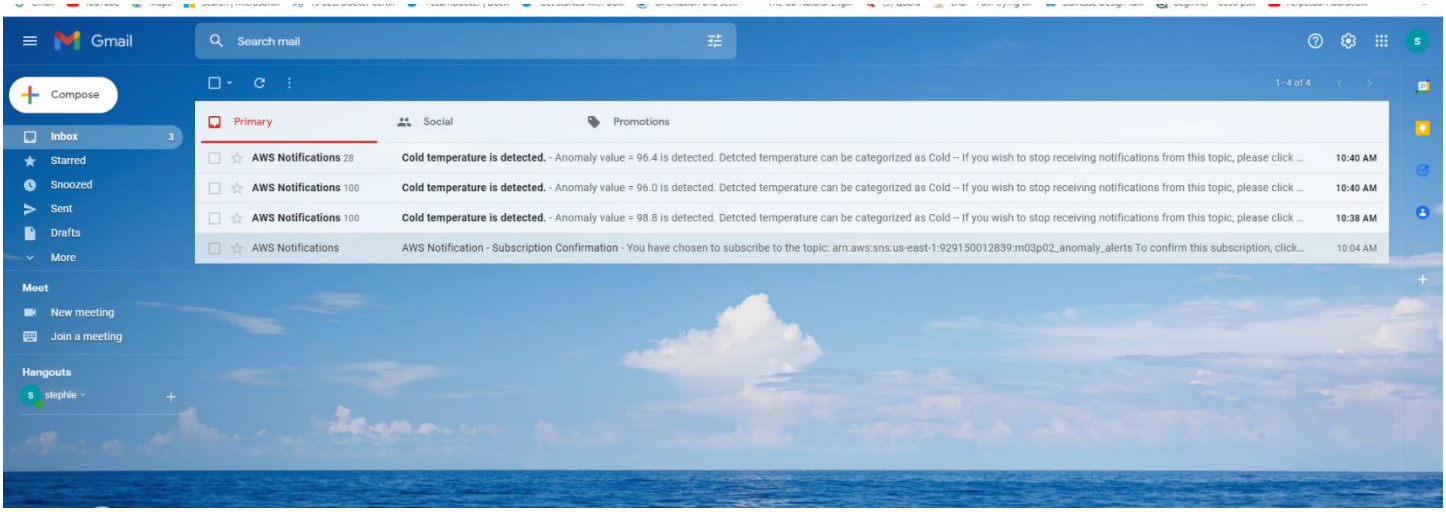
< Screenshot c (5) of Anomaly Data pushed in Dynamodb table for TASK 3>



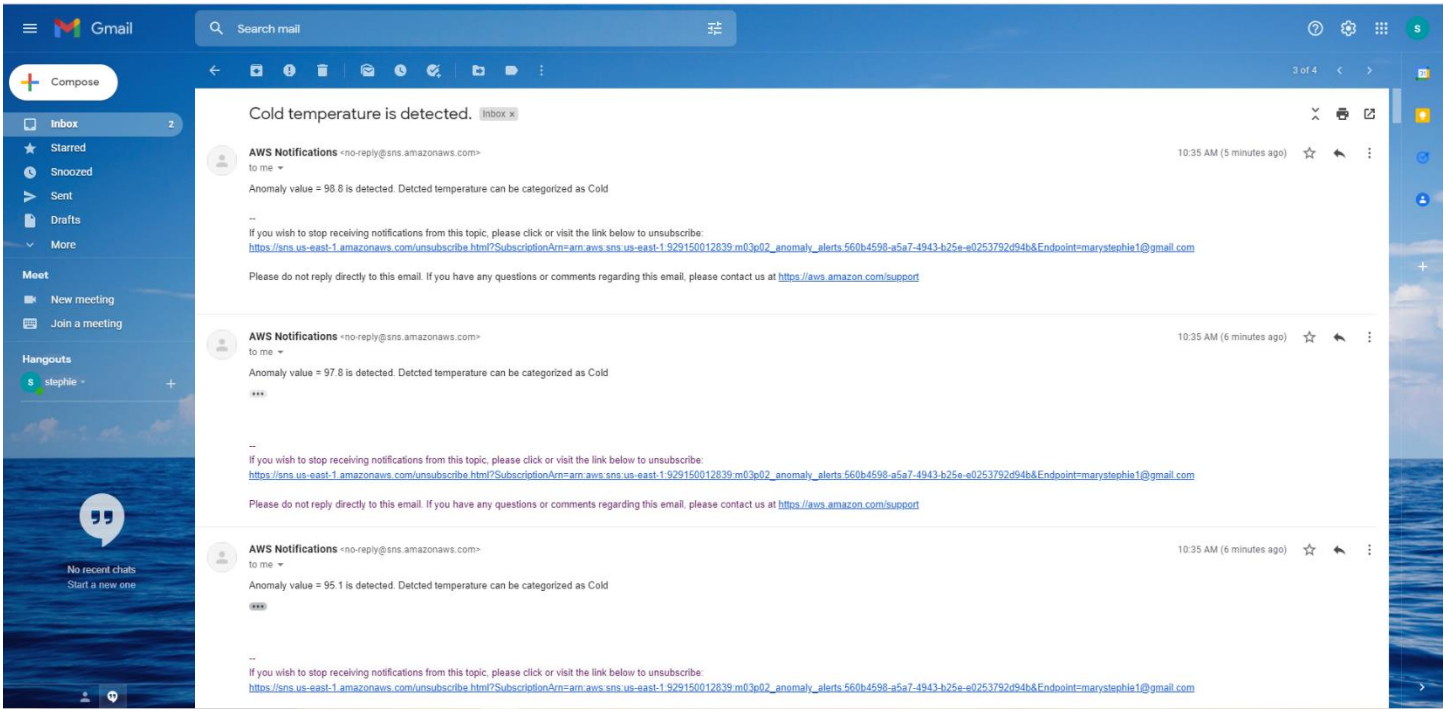
< Screenshot c (5) of Anomaly Data pushed in Dynamodb table for TASK 3>



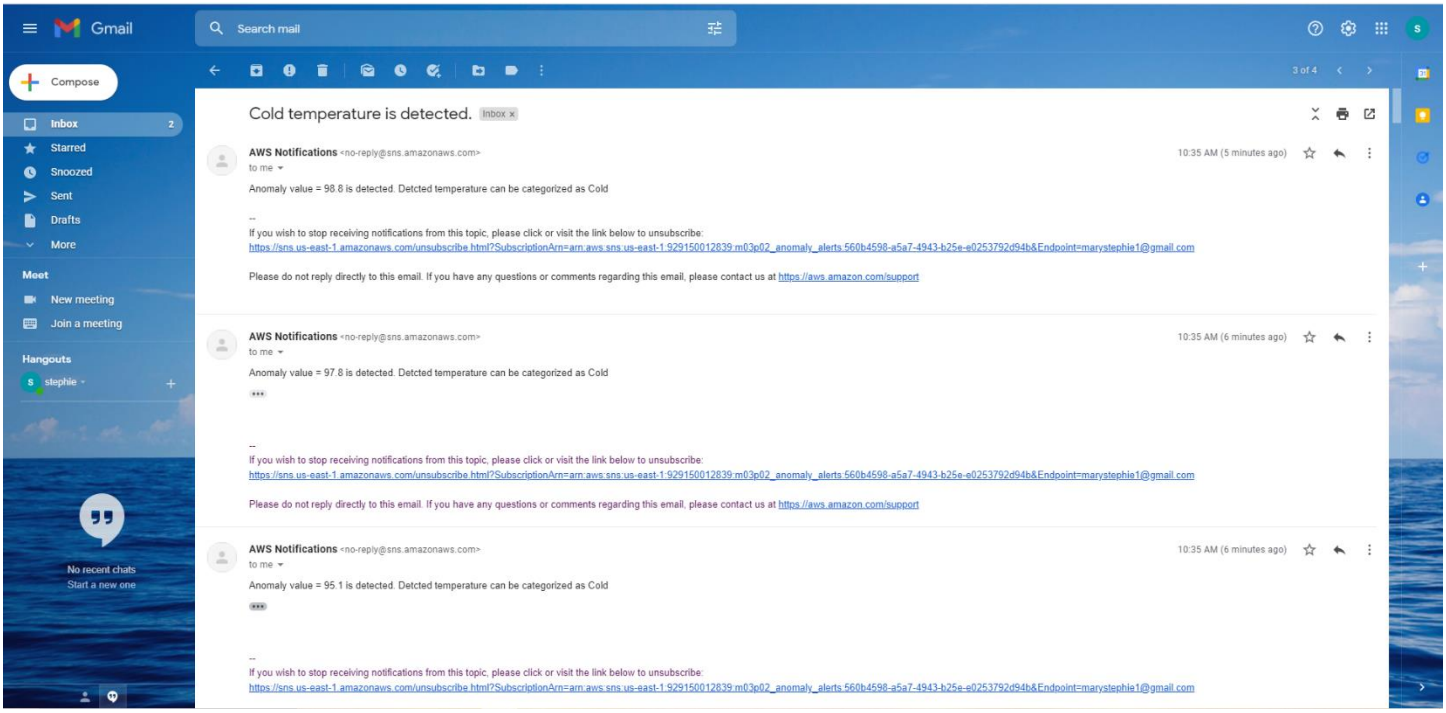
<Screenshot c (6) of SNS notification for TASK 3>



<Screenshot c (6) of SNS notification for TASK 3>



<Screenshot c (6) of SNS notification for TASK 3>



<Screenshot c (6) of SNS notification for TASK 3>