DYNAMO DB

DYNAMODB IS A HOSTED NOSQL DATABASE OFFERED BY AMAZON WEB SERVICES (AWS). IT OFFERS:

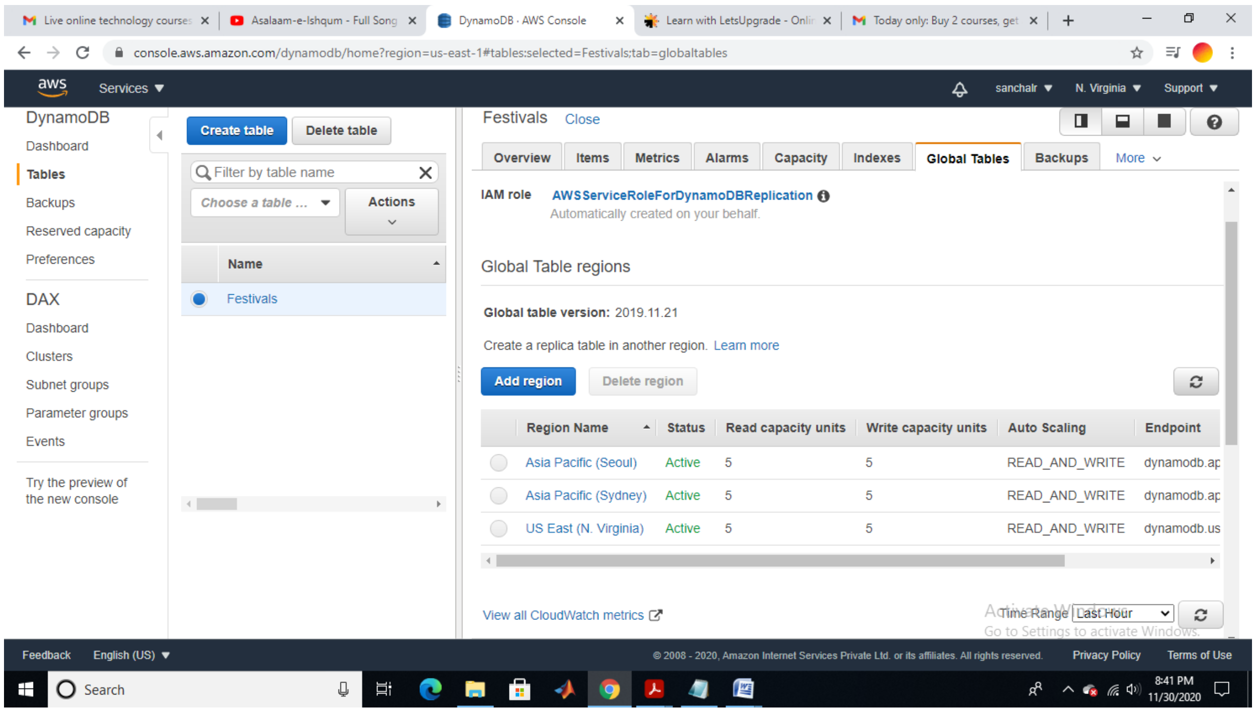
- **RELIABLE PERFORMANCE EVEN AS IT SCALES.**
- **A MANAGED EXPERIENCE, SO YOU WON'T BE SSH-ING INTO SERVERS TO UPGRADE THE CRYPTO LIBRARIES.**
- **A SMALL, SIMPLE API ALLOWING FOR SIMPLE KEY-VALUE ACCESS AS WELL AS MORE ADVANCED QUERY PATTERNS.**

DYNAMODB IS A PARTICULARLY GOOD FIT FOR THE FOLLOWING USE CASES. THEY ARE,

- **APPLICATIONS WITH LARGE AMOUNTS OF DATA AND STRICT LATENCY REQUIREMENTS.**
- **SERVERLESS APPLICATIONS USING AWS LAMBDA.**
- **DATA SETS WITH SIMPLE, KNOWN ACCESS PATTERNS.**

TASK 1: CREATE A DYNAMO DB TABLE WITH MINIMUM TWO DISASTER RECOVERY ZONES AND VERIFY REPLICATION.

TABLE NAMED FESTIVALS IS CREATED.



console.aws.amazon.com/dynamodb/home?region=us-east-1#tables:selected=Festivals;tab=globaltables

Festivals

IAM role AWSServiceRoleForDynamoDBReplication
Automatically created on your behalf.

Global Table regions

Global table version: 2019.11.21

Create a replica table in another region. Learn more

Region Name	Status	Read capacity units	Write capacity units	Auto Scaling	Endpoint
Asia Pacific (Seoul)	Active	5	5	READ_AND_WRITE	dynamodb.ap
Asia Pacific (Sydney)	Active	5	5	READ_AND_WRITE	dynamodb.ap
US East (N. Virginia)	Active	5	5	READ_AND_WRITE	dynamodb.us

View all CloudWatch metrics

Time Range Last Hour

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

8:41 PM 11/30/2020

ABOVE SCREENSHOT SHOWS THE BASE REGION IS N.VIRGINIA.

DISASTER RECOVERY:

DISASTER RECOVERY (DR) IS AN AREA OF SECURITY PLANNING THAT AIMS TO PROTECT AN ORGANIZATION FROM THE EFFECTS OF SIGNIFICANT NEGATIVE EVENTS. HAVING A DISASTER

RECOVERY STRATEGY IN PLACE ENABLES AN ORGANIZATION TO MAINTAIN OR QUICKLY RESUME MISSION-CRITICAL FUNCTIONS FOLLOWING A DISRUPTION.

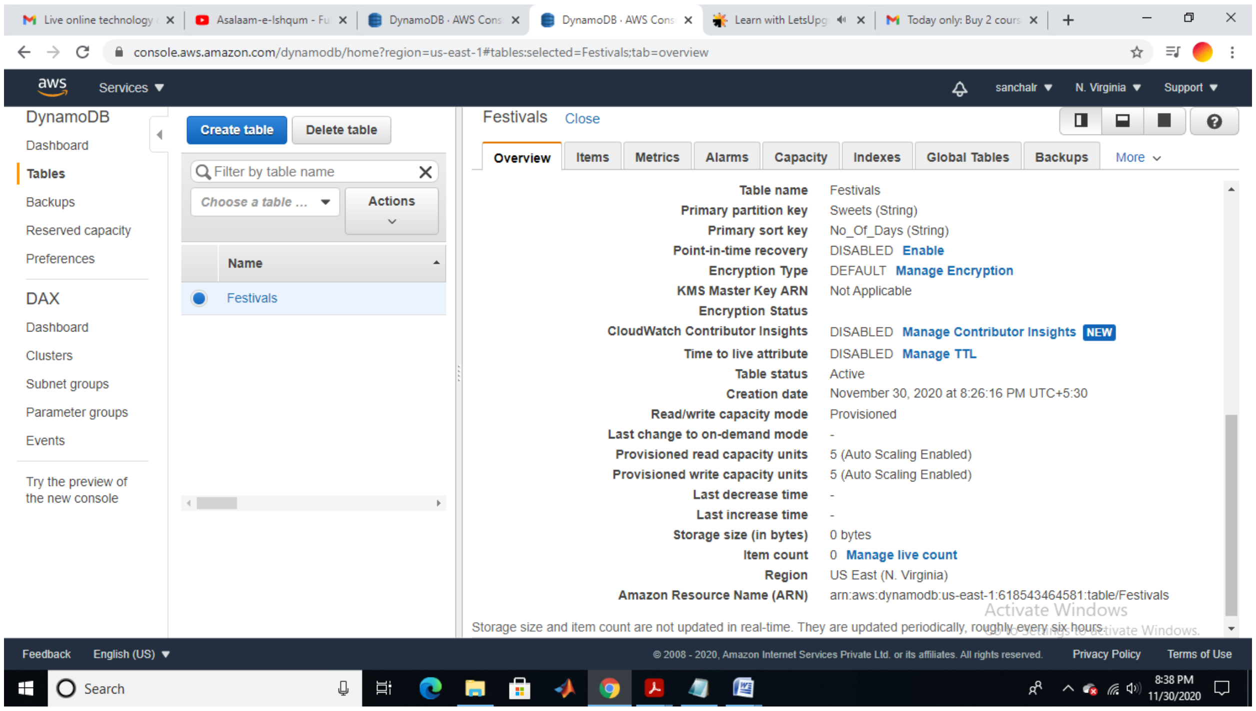


Table name	Festivals
Primary partition key	Sweets (String)
Primary sort key	No_Of_Days (String)
Point-in-time recovery	DISABLED Enable
Encryption Type	DEFAULT Manage Encryption
KMS Master Key ARN	Not Applicable
Encryption Status	
CloudWatch Contributor Insights	DISABLED Manage Contributor Insights NEW
Time to live attribute	DISABLED Manage TTL
Table status	Active
Creation date	November 30, 2020 at 8:26:16 PM UTC+5:30
Read/write capacity mode	Provisioned
Last change to on-demand mode	-
Provisioned read capacity units	5 (Auto Scaling Enabled)
Provisioned write capacity units	5 (Auto Scaling Enabled)
Last decrease time	-
Last increase time	-
Storage size (in bytes)	0 bytes
Item count	0 Manage live count
Region	US East (N. Virginia)
Amazon Resource Name (ARN)	arn:aws:dynamodb:us-east-1:618543464581:table/Festivals

Storage size and item count are not updated in real-time. They are updated periodically, roughly every six hours.

<u>THE FIRST DISASTER RECOVERY REGION IS LOCATED IN SYDNEY.</u>

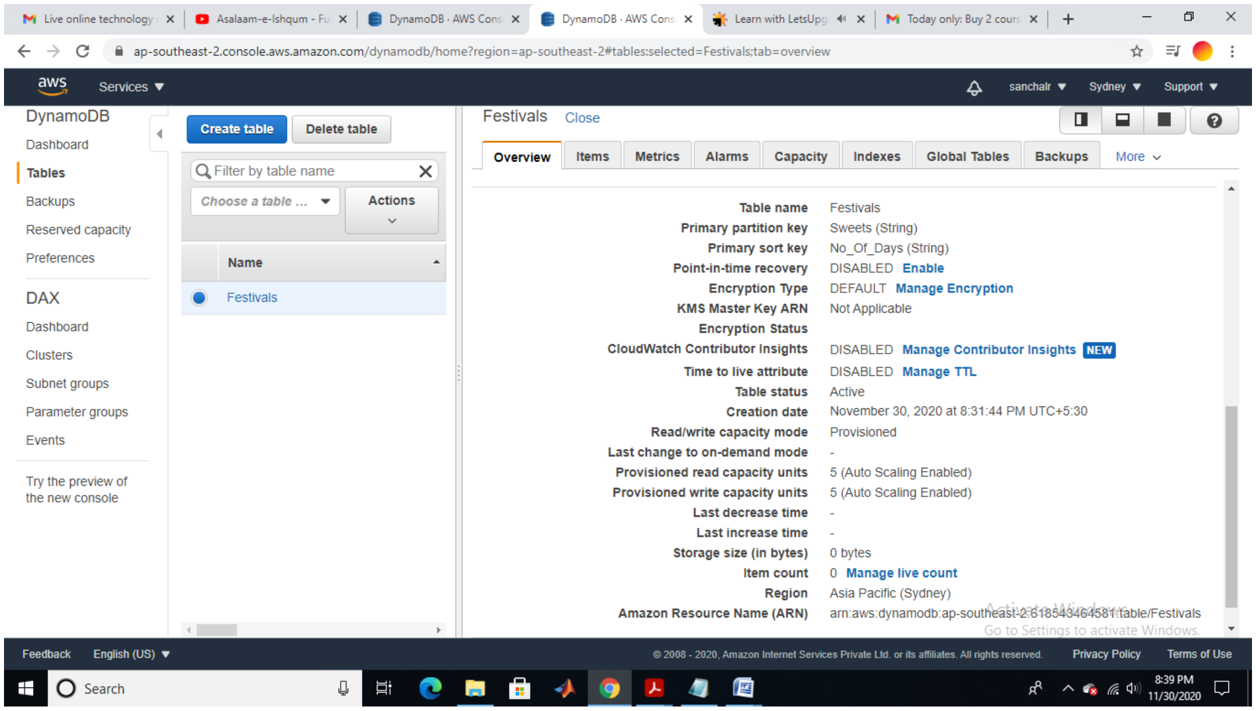
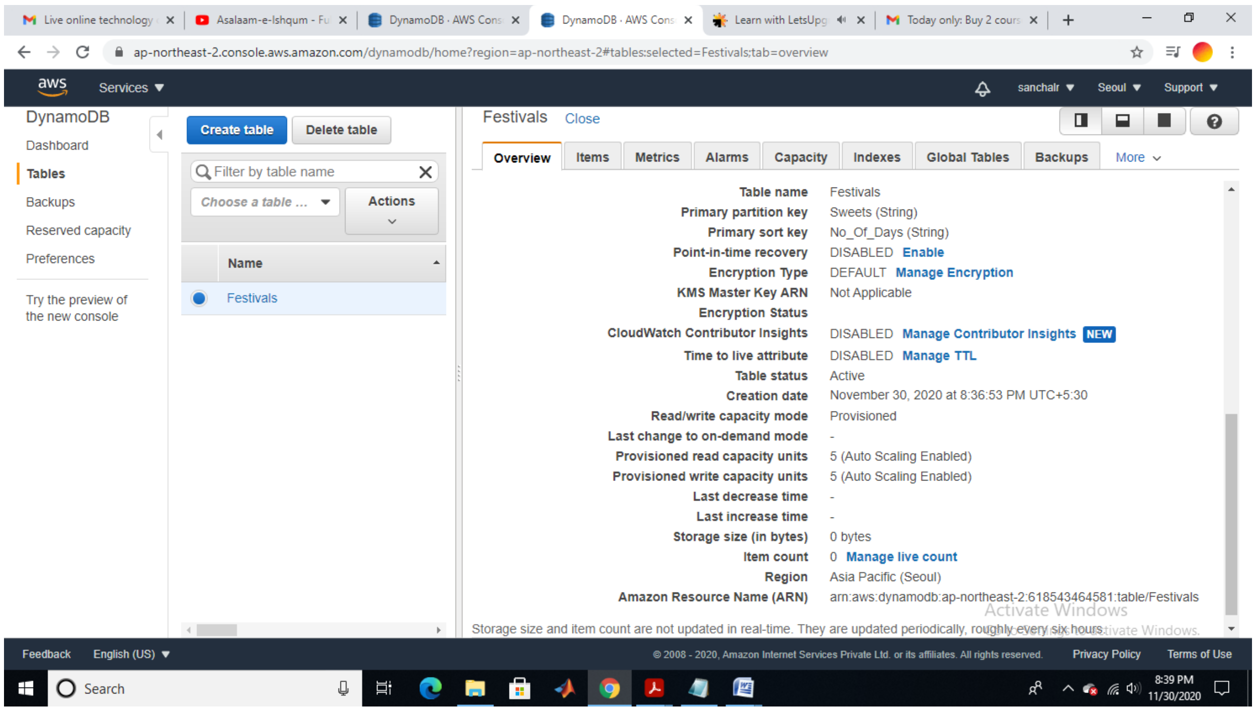


Table name	Festivals
Primary partition key	Sweets (String)
Primary sort key	No_Of_Days (String)
Point-in-time recovery	DISABLED Enable
Encryption Type	DEFAULT Manage Encryption
KMS Master Key ARN	Not Applicable
Encryption Status	
CloudWatch Contributor Insights	DISABLED Manage Contributor Insights NEW
Time to live attribute	DISABLED Manage TTL
Table status	Active
Creation date	November 30, 2020 at 8:31:44 PM UTC+5:30
Read/write capacity mode	Provisioned
Last change to on-demand mode	-
Provisioned read capacity units	5 (Auto Scaling Enabled)
Provisioned write capacity units	5 (Auto Scaling Enabled)
Last decrease time	-
Last increase time	-
Storage size (in bytes)	0 bytes
Item count	0 Manage live count
Region	Asia Pacific (Sydney)
Amazon Resource Name (ARN)	arn:aws:dynamodb:ap-southeast-2:618543464581:table/Festivals

<u>THE SECOND DISASTER RECOVERY REGION IS LOCATED IN SEOUL.</u>



ap-northeast-2.console.aws.amazon.com/dynamodb/home?region=ap-northeast-2#tables:selected=Festivals;tab=overview

Festivals

Table name	Festivals
Primary partition key	Sweets (String)
Primary sort key	No_Of_Days (String)
Point-in-time recovery	DISABLED Enable
Encryption Type	DEFAULT Manage Encryption
KMS Master Key ARN	Not Applicable
Encryption Status	
CloudWatch Contributor Insights	DISABLED Manage Contributor Insights NEW
Time to live attribute	DISABLED Manage TTL
Table status	Active
Creation date	November 30, 2020 at 8:36:53 PM UTC+5:30
Read/write capacity mode	Provisioned
Last change to on-demand mode	-
Provisioned read capacity units	5 (Auto Scaling Enabled)
Provisioned write capacity units	5 (Auto Scaling Enabled)
Last decrease time	-
Last increase time	-
Storage size (in bytes)	0 bytes
Item count	0 Manage live count
Region	Asia Pacific (Seoul)
Amazon Resource Name (ARN)	arn:aws:dynamodb:ap-northeast-2:618543464581:table/Festivals

Storage size and item count are not updated in real-time. They are updated periodically, roughly every six hours.

<u>DISASTER RECOVERY (DR):</u>

A STRATEGIC SECURITY PLANNING MODEL THAT SEEKS TO PROTECT AN ENTERPRISE FROM THE EFFECTS OF NATURAL OR HUMAN-INDUCED DISASTER, SUCH AS A TORNADO OR CYBER ATTACK.
A DR PLAN AIMS TO MAINTAIN CRITICAL FUNCTIONS BEFORE, DURING, AND AFTER A DISASTER EVENT, THEREBY CAUSING MINIMAL DISRUPTION TO BUSINESS CONTINUITY.

<u>BACKUP:</u>

THE COPYING OF DATA INTO A SECONDARY FORM (I.E. ARCHIVE FILE), WHICH CAN BE USED TO RESTORE THE ORIGINAL FILE IN THE EVENT OF A DISASTER EVENT

<u>BELOW SCREENSHOT SHOWS THE ADDING OF ITEMS IN TABLE:</u>

THE ALLOWED DATA TYPES OF ITEMS ARE STRING, BOOLEAN, BYTE, DATE, CALENDAR, LONG, INTEGER, DOUBLE, FLOAT, BIGDECIMAL, BIGINTEGER.

<u>KEYS</u>

THE PRIMARY KEYS SERVE AS THE MEANS OF UNIQUE IDENTIFICATION FOR TABLE ITEMS, AND SECONDARY INDEXES PROVIDE QUERY FLEXIBILITY. DYNAMODB STREAMS RECORD EVENTS BY MODIFYING THE TABLE DATA.

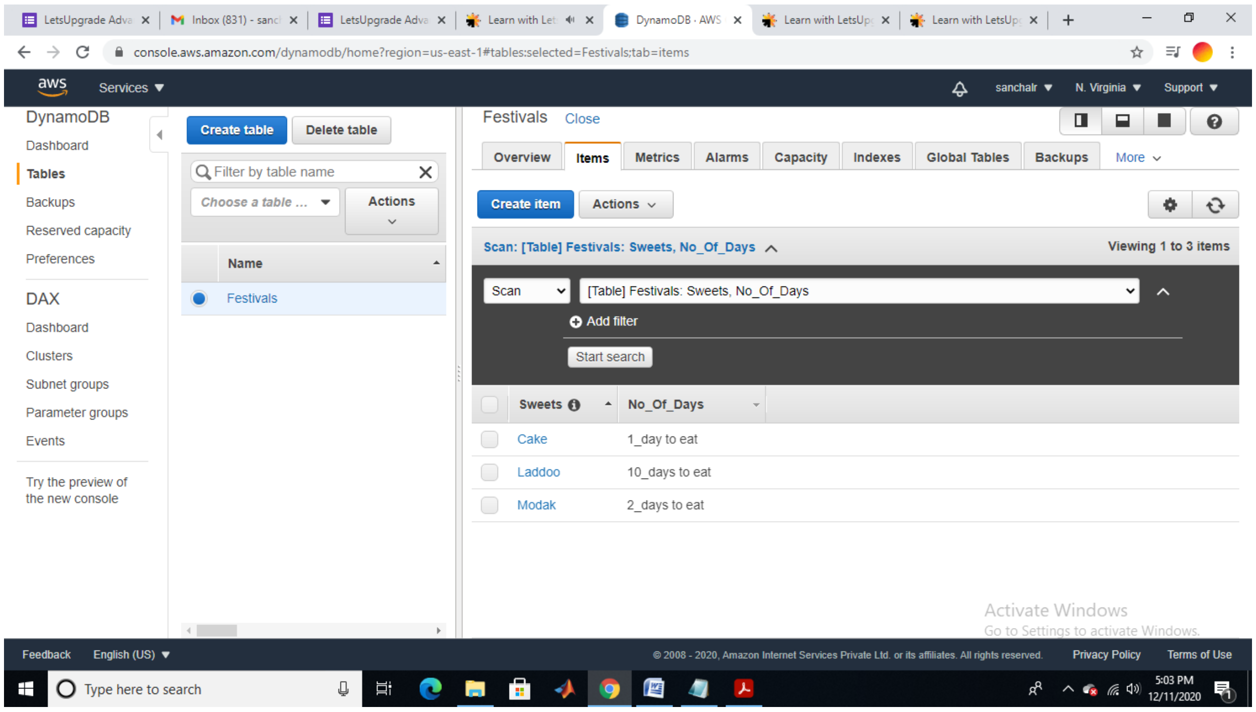
THE TABLE CREATION REQUIRES NOT ONLY SETTING A NAME, BUT ALSO THE PRIMARY KEY; WHICH IDENTIFIES TABLE ITEMS. NO TWO ITEMS SHARE A KEY. DYNAMODB USES TWO TYPES OF PRIMARY KEYS –

- **PARTITION KEY – THIS SIMPLE PRIMARY KEY CONSISTS OF A SINGLE ATTRIBUTE REFERRED TO AS THE "PARTITION KEY." INTERNALLY, DYNAMODB USES THE KEY VALUE AS INPUT FOR A HASH FUNCTION TO DETERMINE STORAGE.**
- **PARTITION KEY AND SORT KEY – THIS KEY, KNOWN AS THE "COMPOSITE PRIMARY KEY", CONSISTS OF TWO ATTRIBUTES.**
    - **THE PARTITION KEY AND**
    - **THE SORT KEY.**

DYNAMODB APPLIES THE FIRST ATTRIBUTE TO A HASH FUNCTION, AND STORES ITEMS WITH THE SAME PARTITION KEY TOGETHER; WITH THEIR ORDER DETERMINED BY THE SORT KEY. ITEMS CAN SHARE PARTITION KEYS, BUT NOT SORT KEYS.

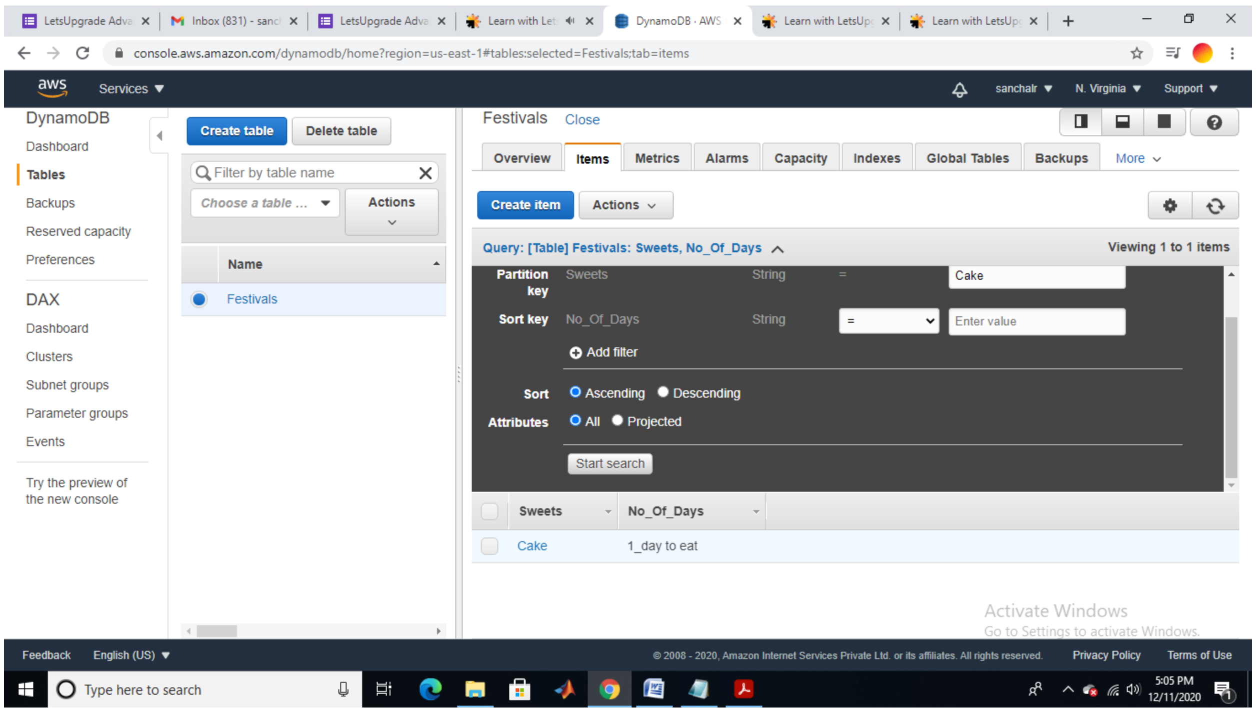
THE PRIMARY KEY ATTRIBUTES ONLY ALLOW SCALAR (SINGLE) VALUES; AND STRING, NUMBER, OR BINARY DATA TYPES. THE NON-KEY ATTRIBUTES DO NOT HAVE THESE CONSTRAINTS.

CAKE, LADDOO, MODAK ARE SWEETS(PARTITION KEY) AND NO_OF_DAYS IS THE SORT KEY.



Scan: [Table] Festivals: Sweets, No_Of_Days — Viewing 1 to 3 items

Sweets	No_Of_Days
Cake	1_day to eat
Laddoo	10_days to eat
Modak	2_days to eat



Festivals Close

Query: [Table] Festivals: Sweets, No_Of_Days

Viewing 1 to 1 items

Partition key: Sweets, String, =, Cake

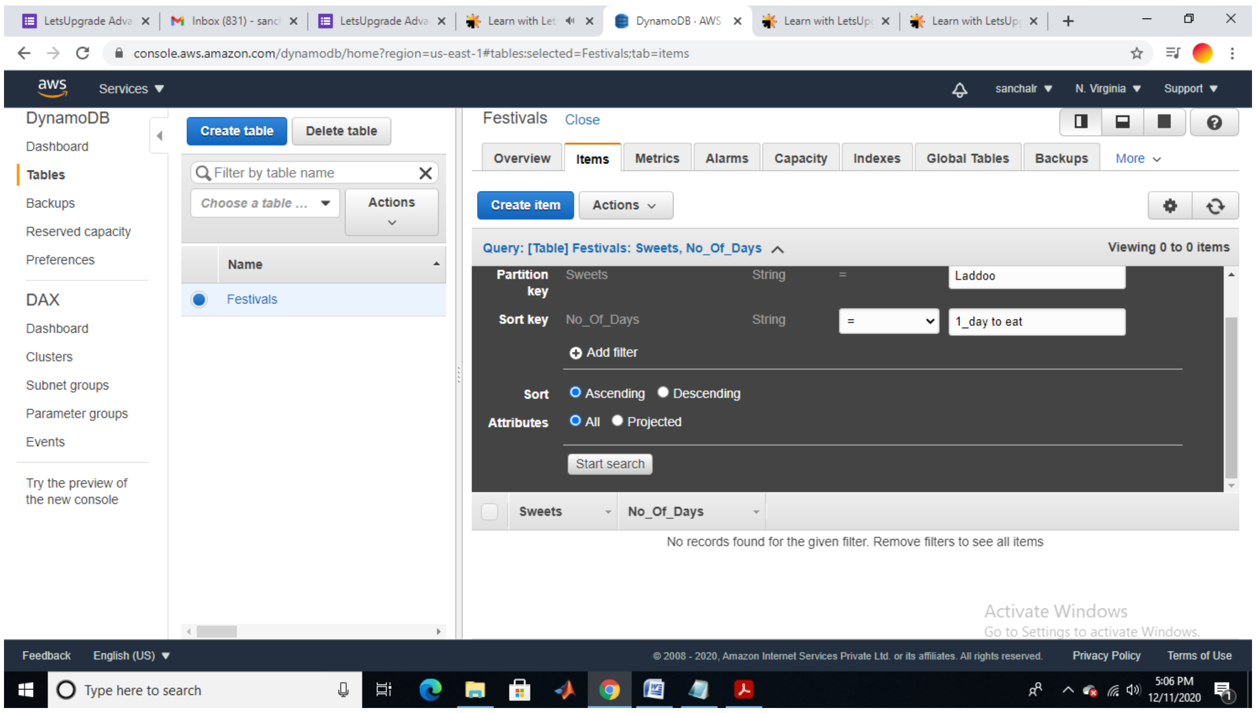
Sort key: No_Of_Days, String, =, Enter value

Sort: Ascending / Descending

Attributes: All / Projected

Sweets	No_Of_Days
Cake	1_day to eat

<u>BELOW SCREENSHOT SHOWS THE QUERY SEARCH OF ITEM BY PARTITION KEY AND SORT KEY. BOTH DETAILS SHOULD MATCH IN ORDER TO GET THE DESIRED RESULT.</u>



Festivals Close

Query: [Table] Festivals: Sweets, No_Of_Days

Viewing 0 to 0 items

Partition key: Sweets, String, =, Laddoo

Sort key: No_Of_Days, String, =, 1_day to eat

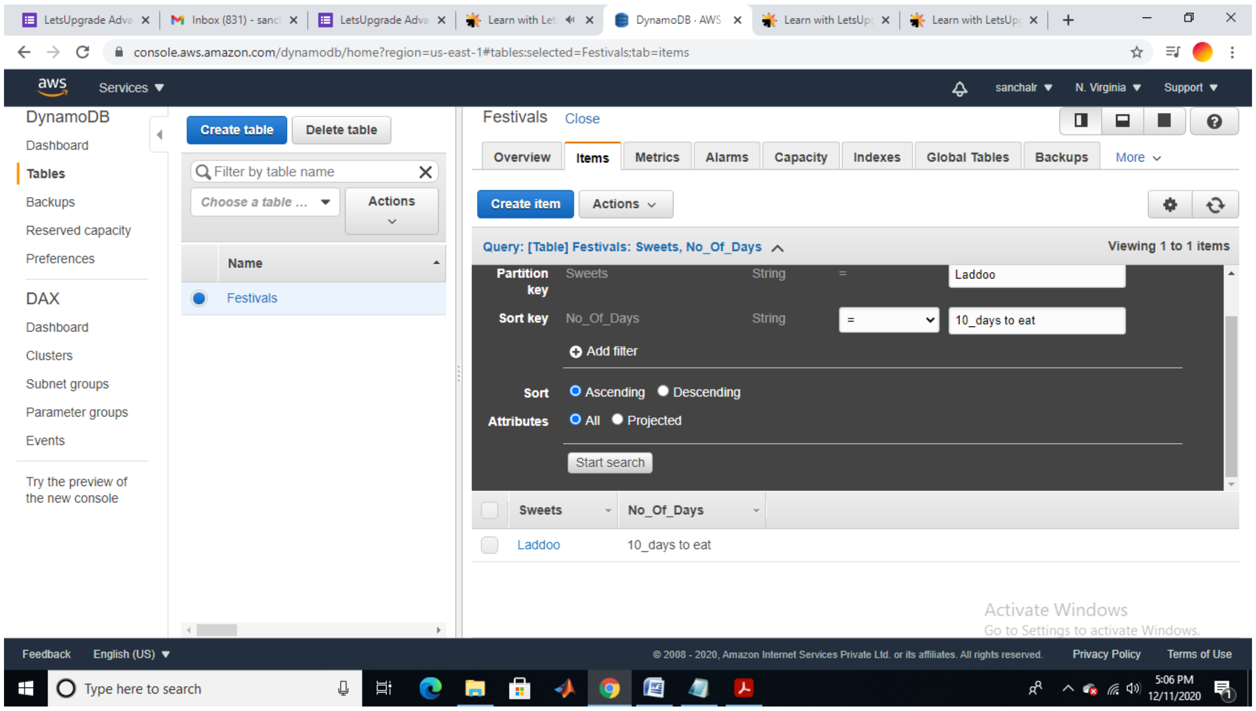
Sort: Ascending / Descending

Attributes: All / Projected

Sweets	No_Of_Days

No records found for the given filter. Remove filters to see all items

<u>BELOW SCREENSHOT HAS THE APPROPRIATE DETAILS MATCHED AND HENCE THE RESULTS IS DISPLAYED ACCORDINGLY.</u>



Query: [Table] Festivals: Sweets, No_Of_Days — Viewing 1 to 1 items

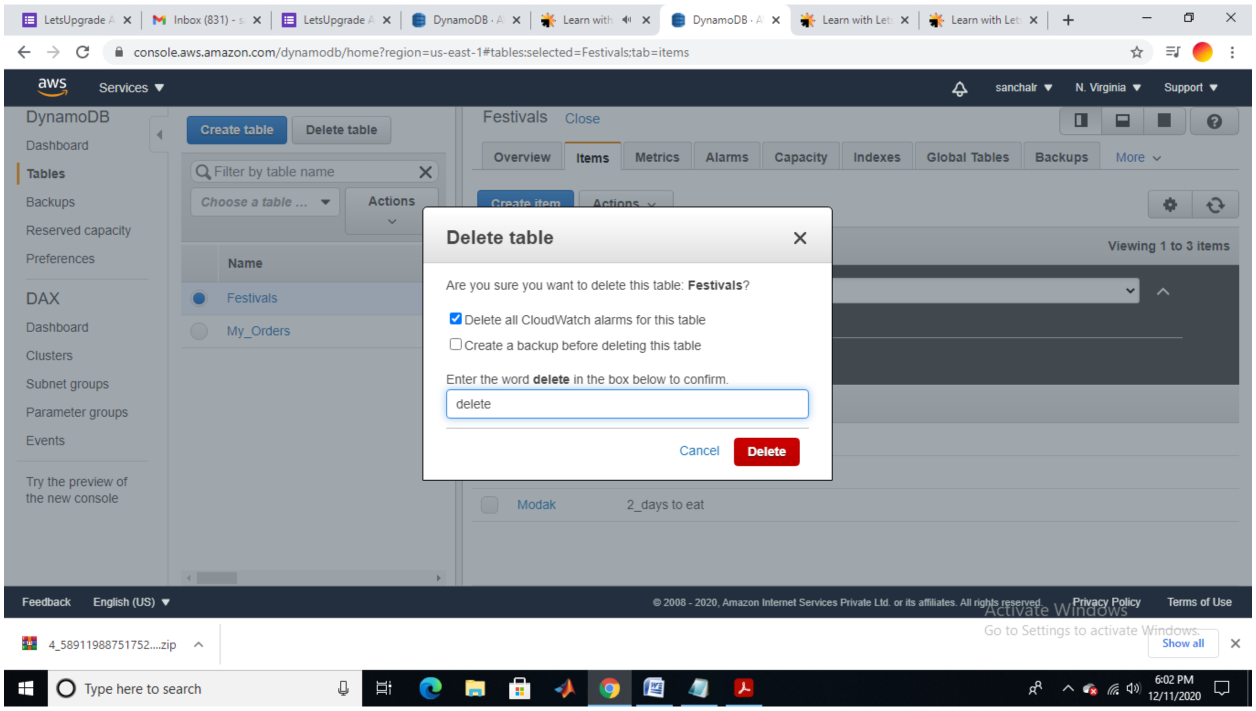
Partition key: Sweets String = Laddoo
Sort key: No_Of_Days String = 10_days to eat
Sort: Ascending / Descending
Attributes: All / Projected

Sweets	No_Of_Days
Laddoo	10_days to eat

DURING DELETION OF TABLE, IF REPLICAS ARE CREATED OF TABLES IN DIFFERENT REGIONS AND IF THE BASE TABLE IS TO BE DELETED, IT WILL SHOW AN ERROR AS REPLICAS ARE CREATED FROM BASE REGION. AND HENCE, BASE TABLE IN BASE REGION WILL BE CREATED WITHIN 24 HOURS.

AND THE REPLICAS AND DATA WILL REMAIN EVEN AFTER BASE REGION TABLE IS DELETED.

IF ALL REPLICAS ARE DELETED FIRST AND THEN THE BASE REGION IS DELETED, THE BASE REGION TABLE WILL GET DELETED.


Delete table

Are you sure you want to delete this table: **Festivals**?

☑ Delete all CloudWatch alarms for this table

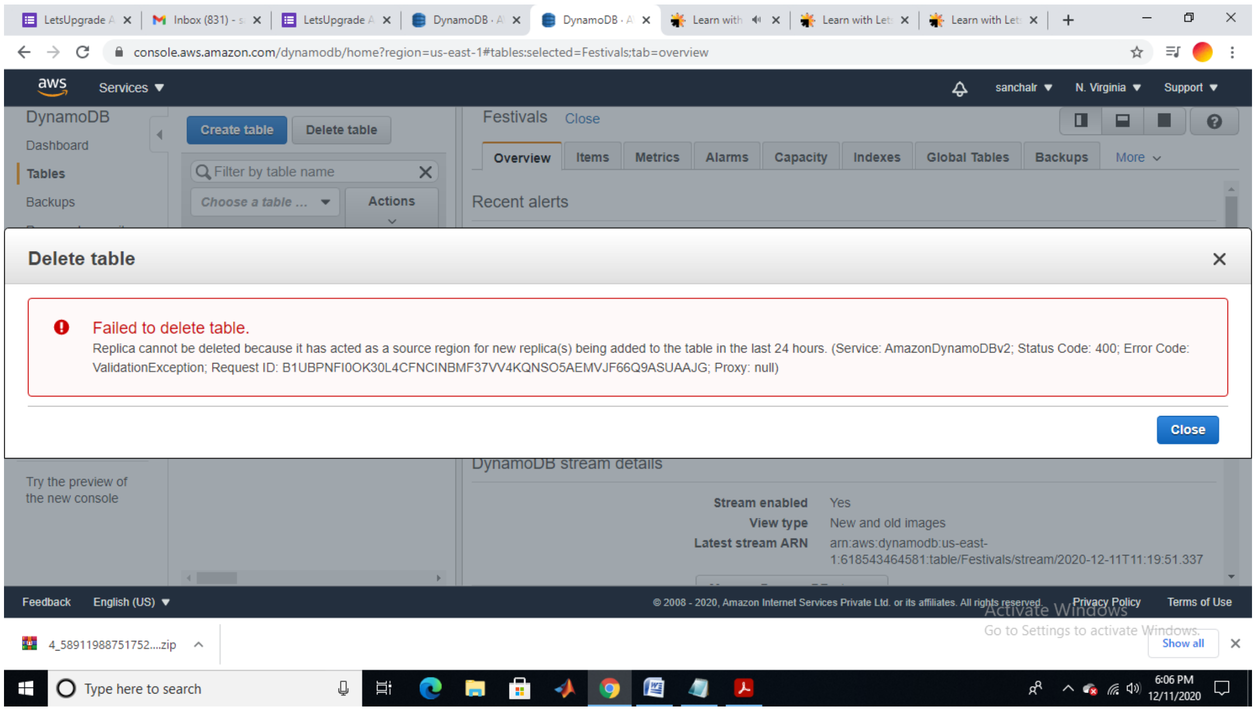
☐ Create a backup before deleting this table

Enter the word **delete** in the box below to confirm.

delete

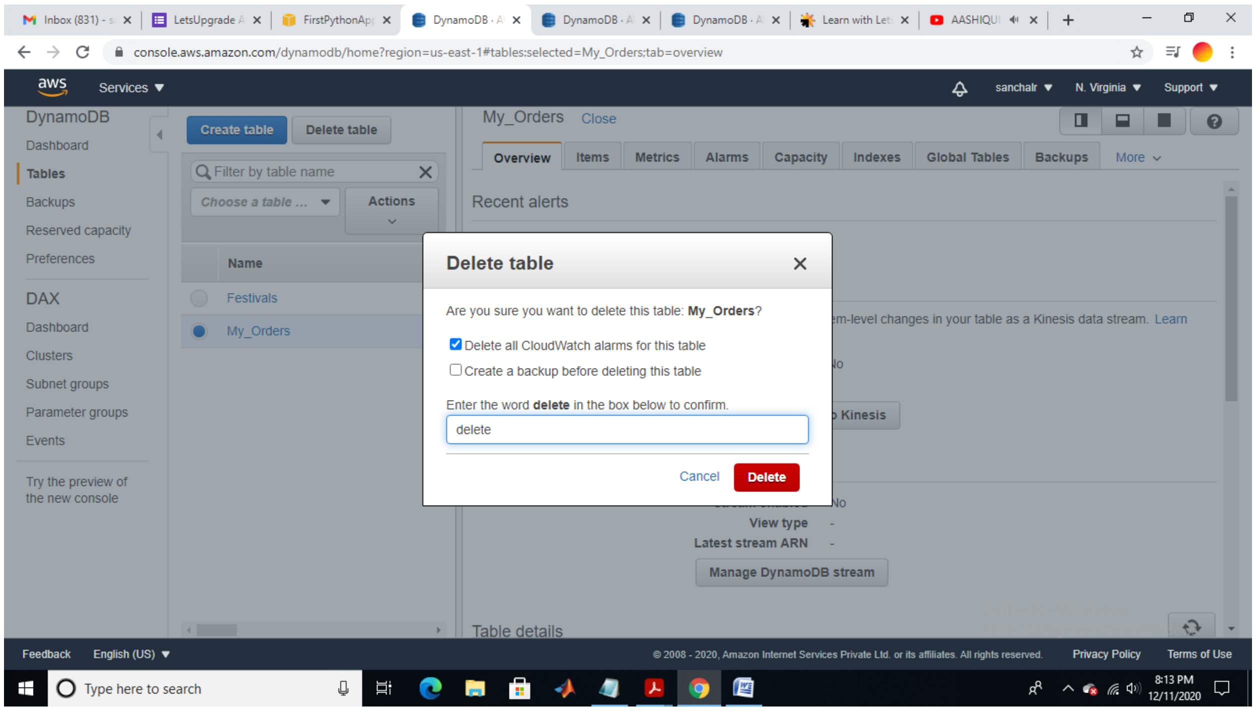
Cancel Delete

BASE REGION IS NOT DELETED DUE TO REPLICAS.


Delete table

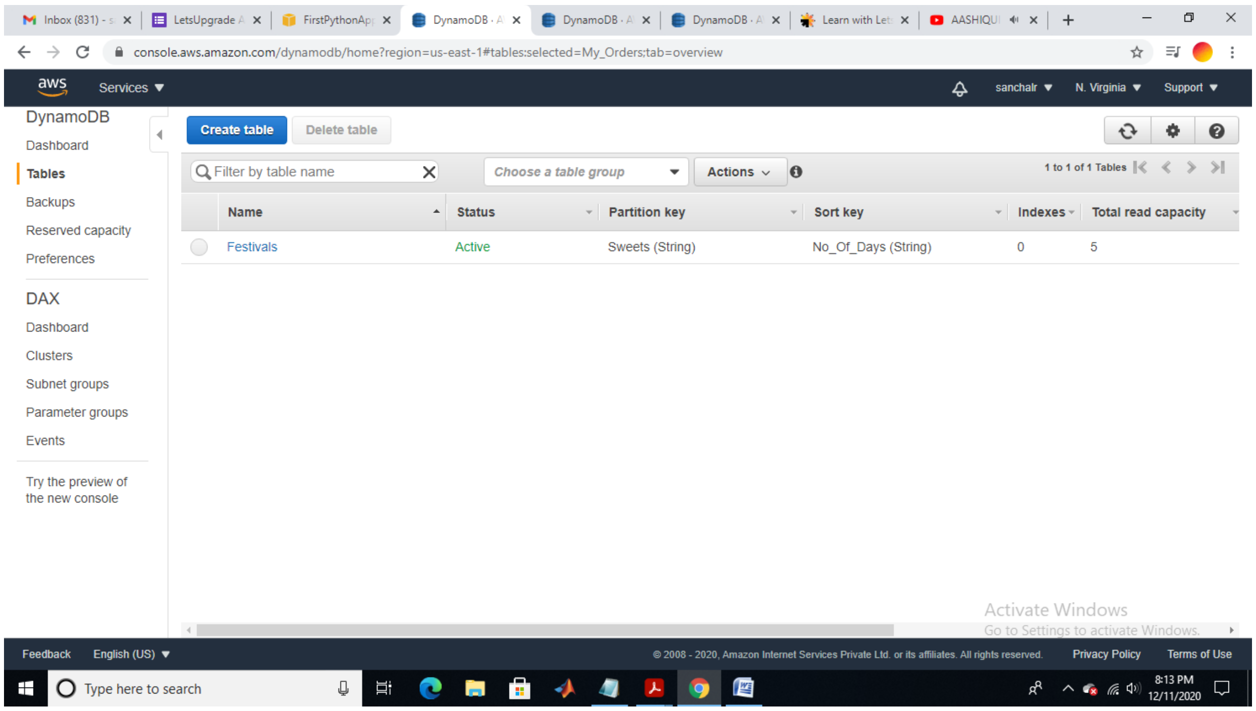
Failed to delete table.
Replica cannot be deleted because it has acted as a source region for new replica(s) being added to the table in the last 24 hours. (Service: AmazonDynamoDBv2; Status Code: 400; Error Code: ValidationException; Request ID: B1UBPNFI0OK30L4CFNCINBMF37VV4KQNSO5AEMVJF66Q9ASUAAJG; Proxy: null)

Close



Delete table
Are you sure you want to delete this table: **My_Orders**?
☑ Delete all CloudWatch alarms for this table
☐ Create a backup before deleting this table
Enter the word **delete** in the box below to confirm.
delete
Cancel
Delete

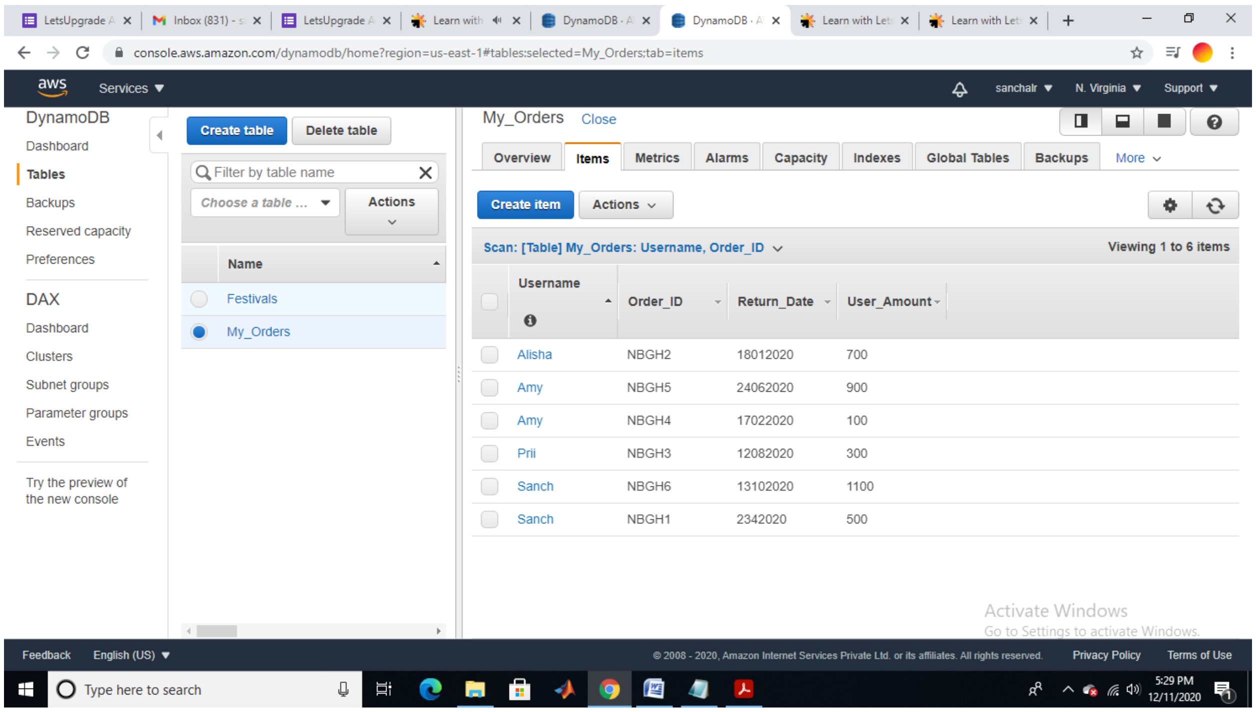
<u>MY-ORDERS TABLE HAS NO REPLICA AND HENCE GETS DELETED IMMEDIATELY AFTER DELETION OPERATION.</u>



Name	Status	Partition key	Sort key	Indexes	Total read capacity
Festivals	Active	Sweets (String)	No_Of_Days (String)	0	5

TASK 2: CREATING A DYNAMO DB TABLE WITH GLOBAL SECONDARY INDEXES AND FETCHING DATA USING GLOBAL SECONDARY INDEXES.

TABLE NAMED MY_ORDERS IS CREATED WITH 6 ITEMS.



Scan: [Table] My_Orders: Username, Order_ID — Viewing 1 to 6 items

Username	Order_ID	Return_Date	User_Amount
Alisha	NBGH2	18012020	700
Amy	NBGH5	24062020	900
Amy	NBGH4	17022020	100
Prii	NBGH3	12082020	300
Sanch	NBGH6	13102020	1100
Sanch	NBGH1	2342020	500

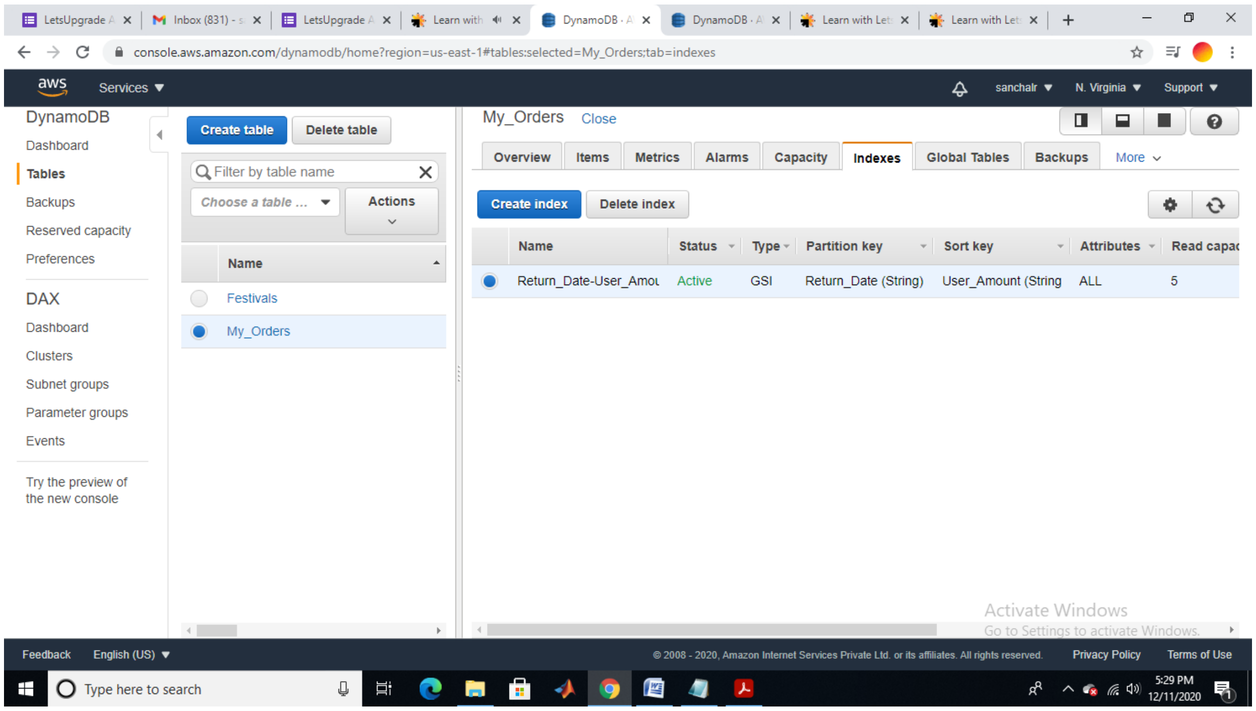
SECONDARY INDEXES

THESE INDEXES ALLOW YOU TO QUERY TABLE DATA WITH AN ALTERNATE KEY. THOUGH DYNAMODB DOES NOT FORCE THEIR USE, THEY OPTIMIZE QUERYING.

DYNAMODB USES TWO TYPES OF SECONDARY INDEXES –

- **GLOBAL SECONDARY INDEX – THIS INDEX POSSESSES PARTITION AND SORT KEYS, WHICH CAN DIFFER FROM TABLE KEYS.**
- **LOCAL SECONDARY INDEX – THIS INDEX POSSESSES A PARTITION KEY IDENTICAL TO THE TABLE , HOWEVER, ITS SORT KEY DIFFERS.**

EACH TABLE IN DYNAMODB CAN HAVE UP TO 20 GLOBAL SECONDARY INDEXES (DEFAULT QUOTA) AND 5 LOCAL SECONDARY INDEXES.

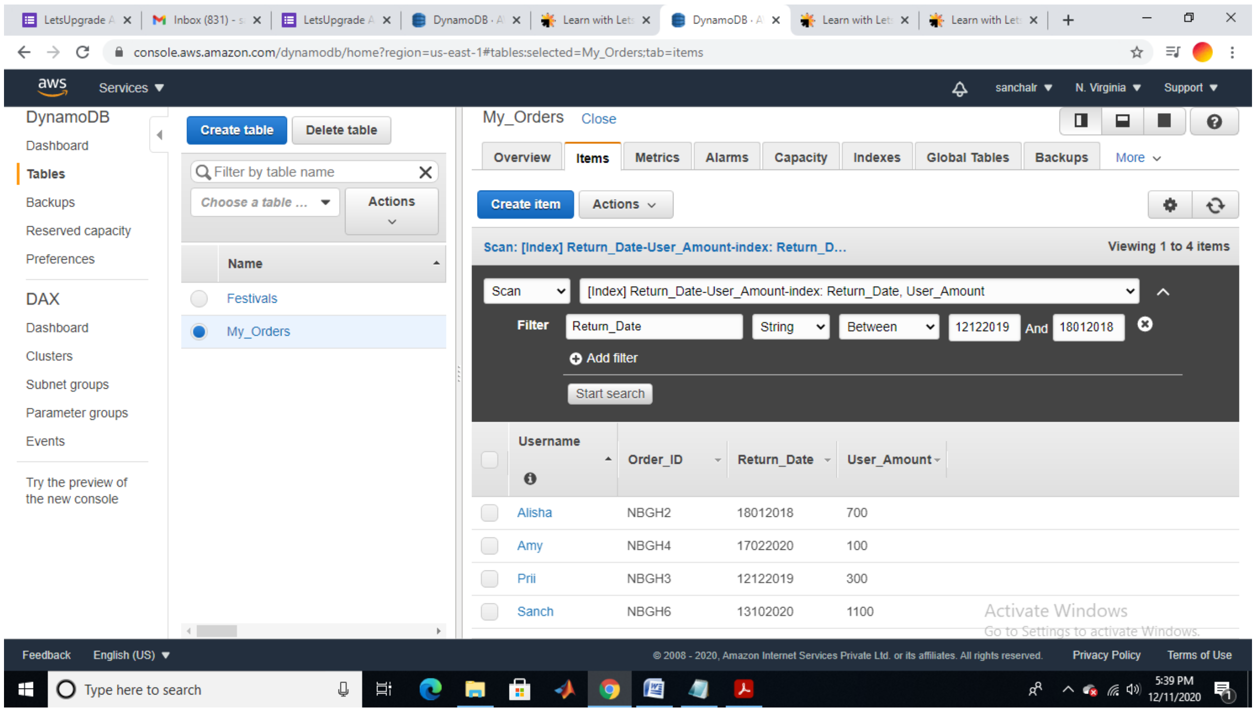


console.aws.amazon.com/dynamodb/home?region=us-east-1#tables:selected=My_Orders;tab=indexes

My_Orders Close

Name	Status	Type	Partition key	Sort key	Attributes	Read capac
Return_Date-User_Amou	Active	GSI	Return_Date (String)	User_Amount (String	ALL	5

BY USING SCAN FUNCTION, USE PARTITION KEY AND SORT KEY TO FETCH DESIRED RESULT.



console.aws.amazon.com/dynamodb/home?region=us-east-1#tables:selected=My_Orders;tab=items

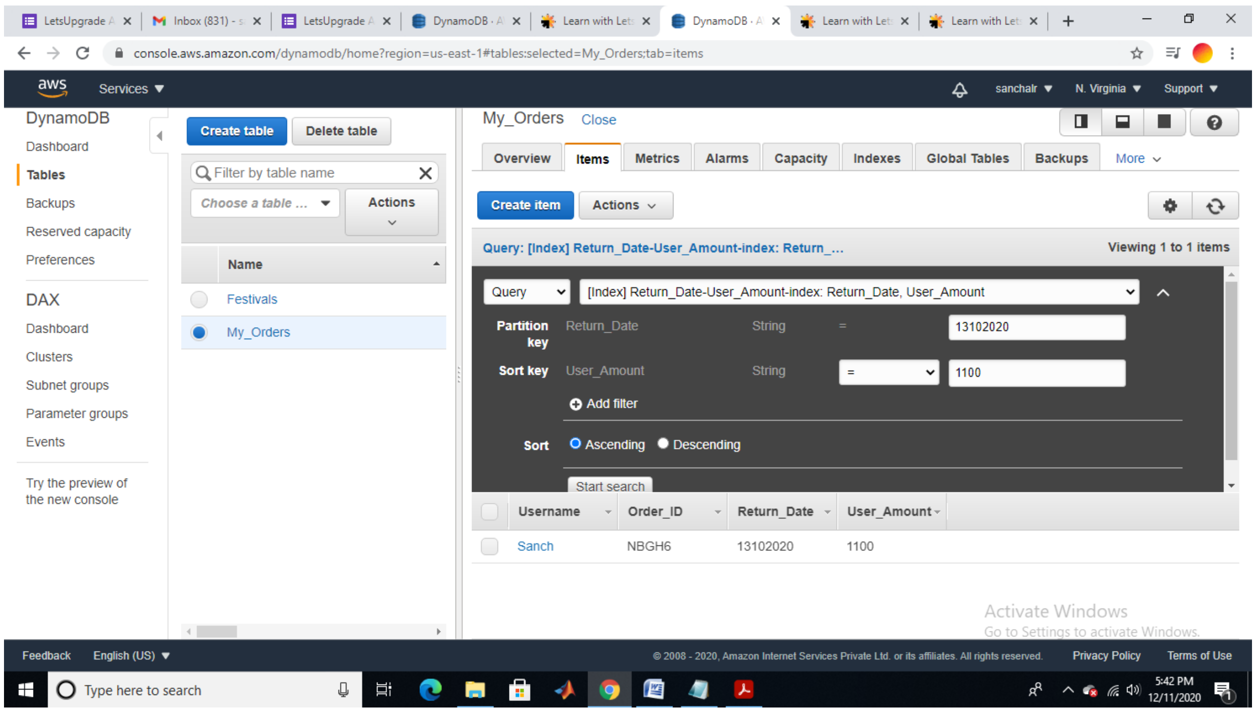
My_Orders Close

Scan: [Index] Return_Date-User_Amount-index: Return_D... Viewing 1 to 4 items

Scan [Index] Return_Date-User_Amount-index: Return_Date, User_Amount

Filter Return_Date String Between 12122019 And 18012018

Username	Order_ID	Return_Date	User_Amount
Alisha	NBGH2	18012018	700
Amy	NBGH4	17022020	100
Prii	NBGH3	12122019	300
Sanch	NBGH6	13102020	1100


console.aws.amazon.com/dynamodb/home?region=us-east-1#tables:selected=My_Orders;tab=items

My_Orders

Query: [Index] Return_Date-User_Amount-index: Return_... Viewing 1 to 1 items

Query [Index] Return_Date-User_Amount-index: Return_Date, User_Amount

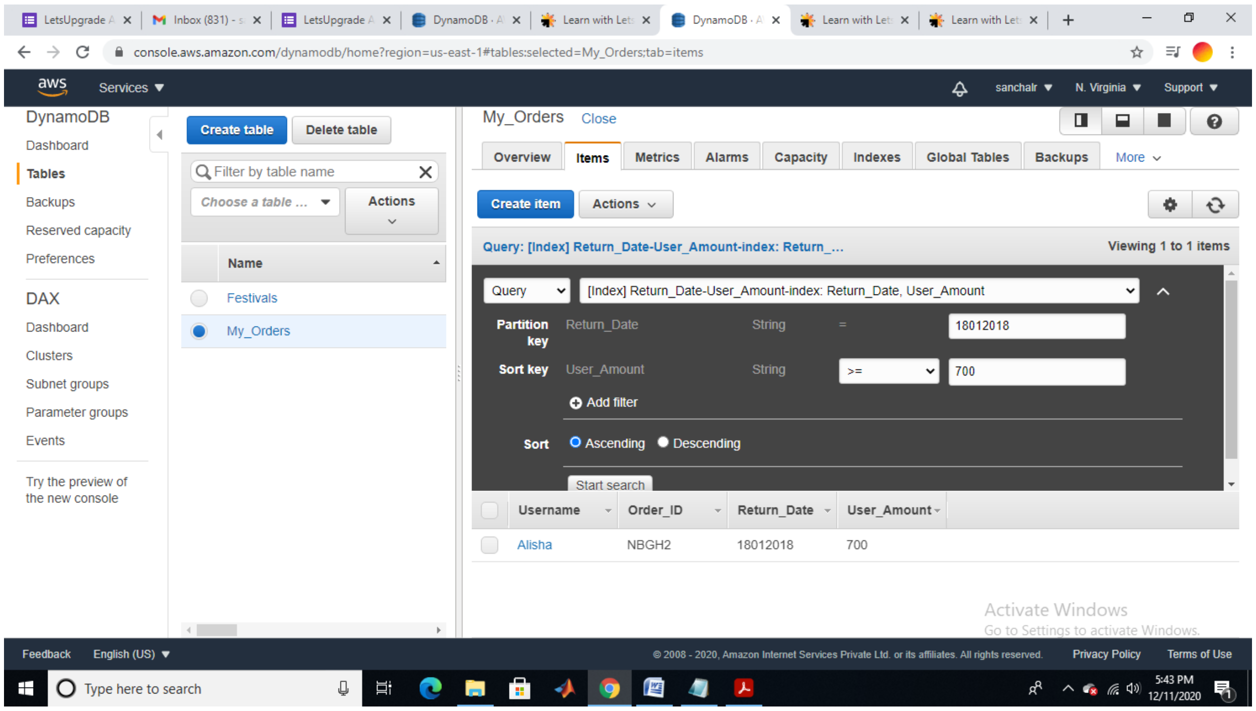
Partition key Return_Date String = 13102020

Sort key User_Amount String = 1100

Sort [✓ Ascending] Descending

Username	Order_ID	Return_Date	User_Amount
Sanch	NBGH6	13102020	1100

BELOW SCREENSHOT SHOWS PARTITION KEY AND SORT KEY SELECTION AND RESULT.


console.aws.amazon.com/dynamodb/home?region=us-east-1#tables:selected=My_Orders;tab=items

My_Orders

Query: [Index] Return_Date-User_Amount-index: Return_... Viewing 1 to 1 items

Query [Index] Return_Date-User_Amount-index: Return_Date, User_Amount

Partition key Return_Date String = 18012018

Sort key User_Amount String >= 700

Sort [✓ Ascending] Descending

Username	Order_ID	Return_Date	User_Amount
Alisha	NBGH2	18012018	700

TASK 3: DEPLOYING A PYTHON APPLICATION IN ELASTIC BEANSTALK.

ELASTIC BEANSTALK MAKES IT EVEN EASIER FOR DEVELOPERS TO QUICKLY DEPLOY AND MANAGE APPLICATIONS IN THE AWS CLOUD. DEVELOPERS SIMPLY UPLOAD THEIR APPLICATION, AND ELASTIC BEANSTALK AUTOMATICALLY HANDLES THE DEPLOYMENT DETAILS OF CAPACITY PROVISIONING, LOAD BALANCING, AUTO-SCALING, AND APPLICATION HEALTH MONITORING.

ADVANTAGES ARE:

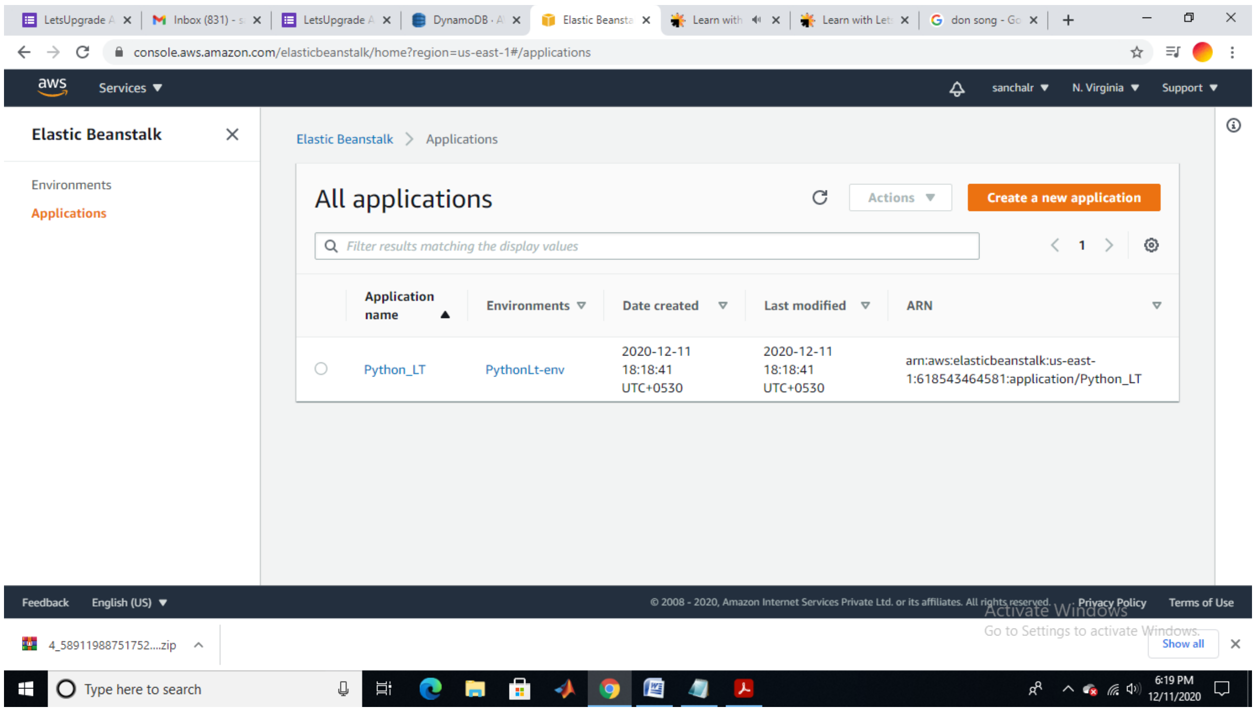
- **LIGHTNING FAST CONFIGURATION WITH AUTOMATION.**
- **POWERFUL CUSTOMIZATION.**
- **PRICE AND FLEXIBILITY.**

DRAWBACKS ARE:

- **UNRELIABLE DEPLOYMENT.**
- **DEPLOYMENT SPEED.**
- **STACK UPGRADES.**

AN AWS ELASTIC BEANSTALK ENVIRONMENT IS A COLLECTION OF AWS RESOURCES RUNNING AN APPLICATION VERSION. YOU CAN DEPLOY MULTIPLE ENVIRONMENTS WHEN YOU NEED TO RUN MULTIPLE VERSIONS OF AN APPLICATION. FOR EXAMPLE, YOU MIGHT HAVE DEVELOPMENT, INTEGRATION, AND PRODUCTION ENVIRONMENTS.

PYTHON ENVIRONMENT IS CREATED NAMED PYTHON_LT.

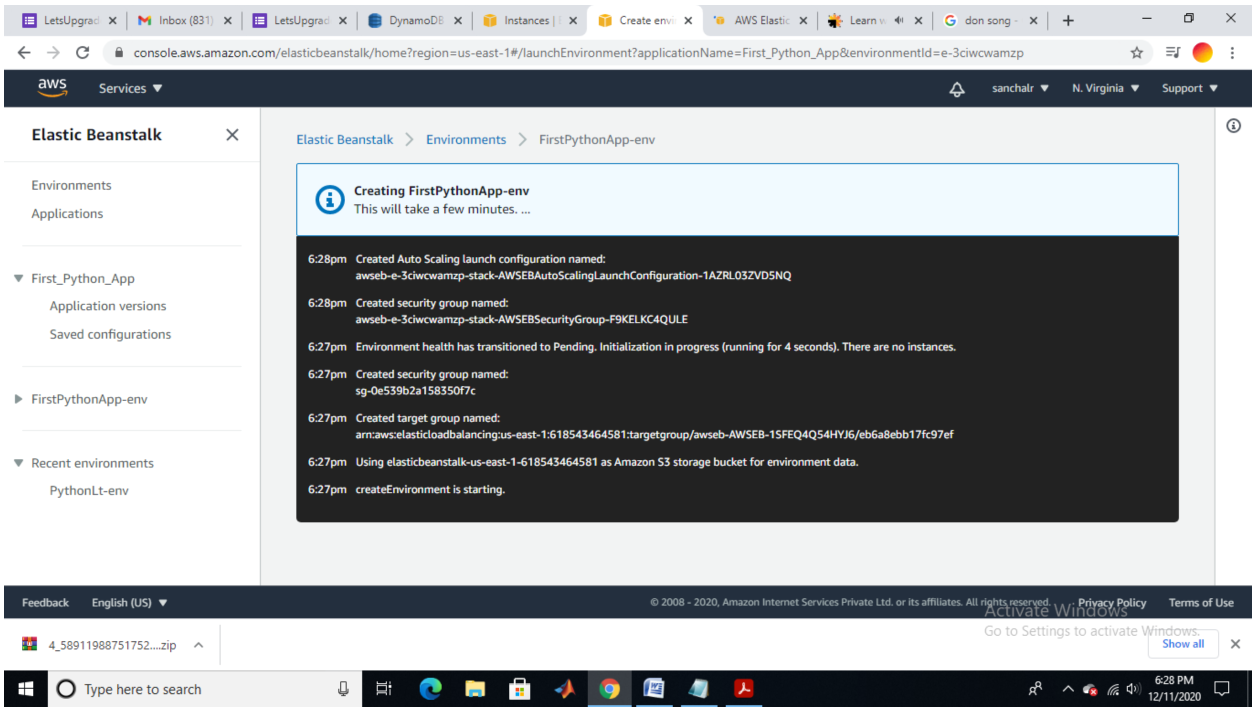


All applications

Application name	Environments	Date created	Last modified	ARN
Python_LT	PythonLt-env	2020-12-11 18:18:41 UTC+0530	2020-12-11 18:18:41 UTC+0530	arn:aws:elasticbeanstalk:us-east-1:618543464581:application/Python_LT

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

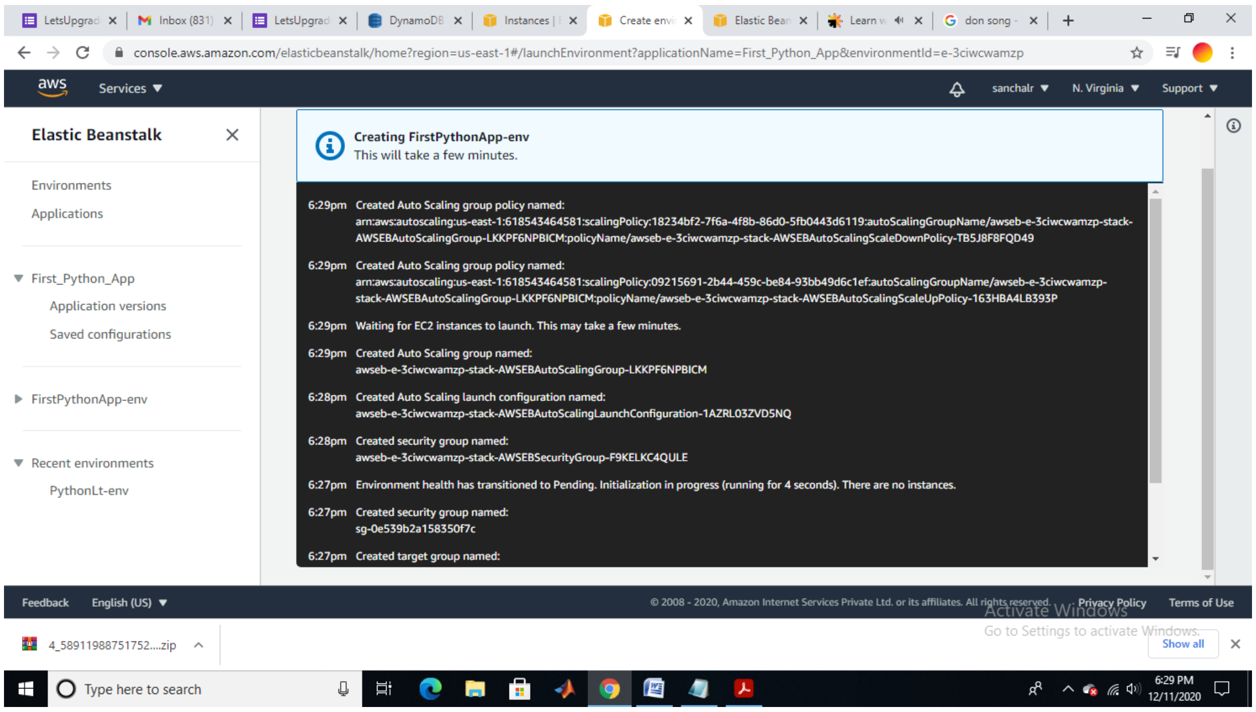
<u>BELOW SCREENSHOT SHOWS THE CREATION ORDERS OF ALL SERVICES REQUIRED TO CREATE THESE ENVIRONMENTS.</u>



console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/launchEnvironment?applicationName=First_Python_App&environmentId=e-3ciwcwamzp

Creating FirstPythonApp-env
This will take a few minutes. ...

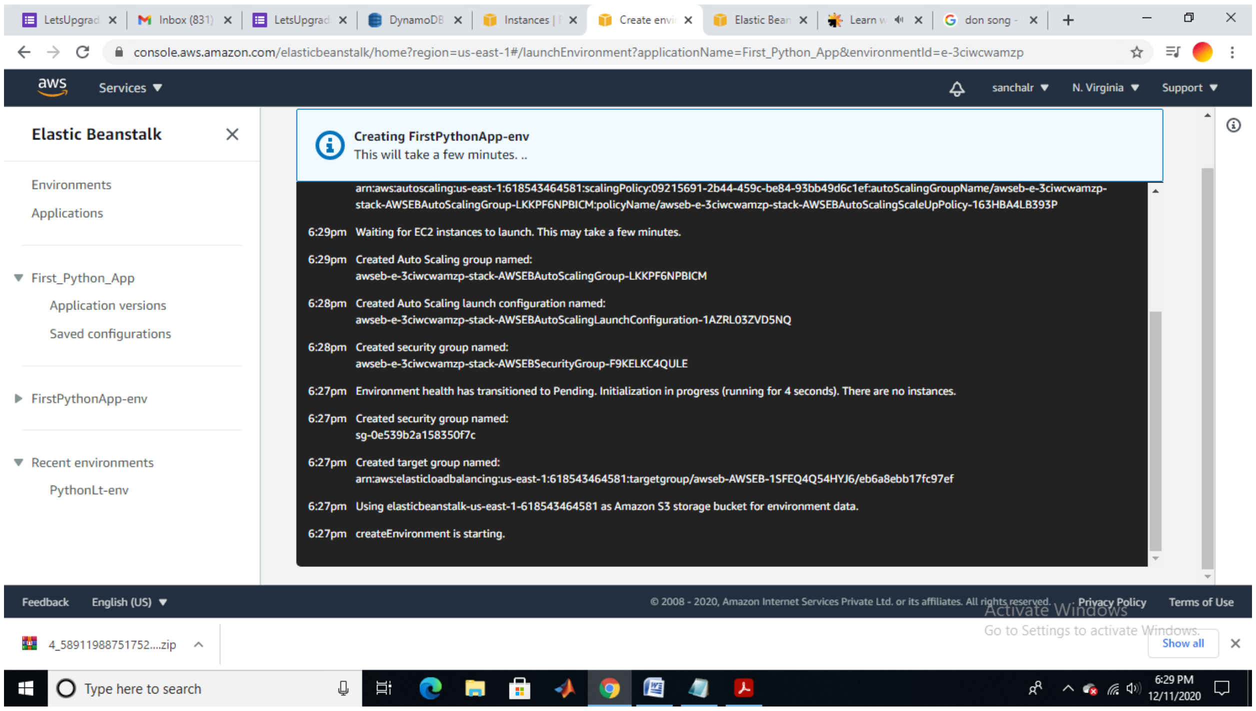
```
6:28pm  Created Auto Scaling launch configuration named:
        awseb-e-3ciwcwamzp-stack-AWSEBAutoScalingLaunchConfiguration-1AZRL03ZVD5NQ
6:28pm  Created security group named:
        awseb-e-3ciwcwamzp-stack-AWSEBSecurityGroup-F9KELKC4QULE
6:27pm  Environment health has transitioned to Pending. Initialization in progress (running for 4 seconds). There are no instances.
6:27pm  Created security group named:
        sg-0e539b2a158350f7c
6:27pm  Created target group named:
        arn:aws:elasticloadbalancing:us-east-1:618543464581:targetgroup/awseb-AWSEB-1SFEQ4Q54HYJ6/eb6a8ebb17fc97ef
6:27pm  Using elasticbeanstalk-us-east-1-618543464581 as Amazon S3 storage bucket for environment data.
6:27pm  createEnvironment is starting.
```



console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/launchEnvironment?applicationName=First_Python_App&environmentId=e-3ciwcwamzp

Creating FirstPythonApp-env
This will take a few minutes.

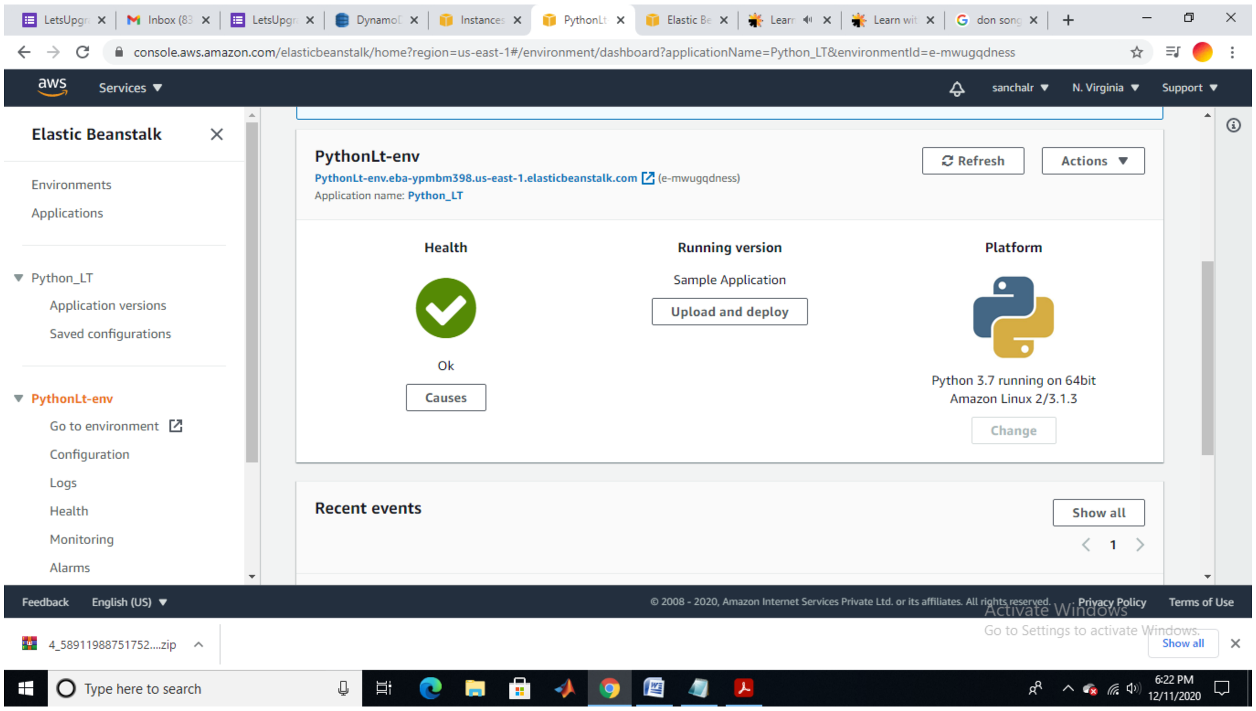
```
6:29pm  Created Auto Scaling group policy named:
        arn:aws:autoscaling:us-east-1:618543464581:scalingPolicy:18234bf2-7f6a-4f8b-86d0-5fb0443d6119:autoScalingGroupName/awseb-e-3ciwcwamzp-stack-
        AWSEBAutoScalingGroup-LKKPF6NPBICM:policyName/awseb-e-3ciwcwamzp-stack-AWSEBAutoScalingScaleDownPolicy-TB5J8F8FQD49
6:29pm  Created Auto Scaling group policy named:
        arn:aws:autoscaling:us-east-1:618543464581:scalingPolicy:09215691-2b44-459c-be84-93bb49d6c1ef:autoScalingGroupName/awseb-e-3ciwcwamzp-
        stack-AWSEBAutoScalingGroup-LKKPF6NPBICM:policyName/awseb-e-3ciwcwamzp-stack-AWSEBAutoScalingScaleUpPolicy-163HBA4LB393P
6:29pm  Waiting for EC2 instances to launch. This may take a few minutes.
6:29pm  Created Auto Scaling group named:
        awseb-e-3ciwcwamzp-stack-AWSEBAutoScalingGroup-LKKPF6NPBICM
6:28pm  Created Auto Scaling launch configuration named:
        awseb-e-3ciwcwamzp-stack-AWSEBAutoScalingLaunchConfiguration-1AZRL03ZVD5NQ
6:28pm  Created security group named:
        awseb-e-3ciwcwamzp-stack-AWSEBSecurityGroup-F9KELKC4QULE
6:27pm  Environment health has transitioned to Pending. Initialization in progress (running for 4 seconds). There are no instances.
6:27pm  Created security group named:
        sg-0e539b2a158350f7c
6:27pm  Created target group named:
```



Creating FirstPythonApp-env
This will take a few minutes. ..

```
        arn:aws:autoscaling:us-east-1:618543464581:scalingPolicy:09215691-2b44-459c-be84-93bb49d6c1ef:autoScalingGroupName/awseb-e-3ciwcwamzp-
        stack-AWSEBAutoScalingGroup-LKKPF6NPBICM:policyName/awseb-e-3ciwcwamzp-stack-AWSEBAutoScalingScaleUpPolicy-163HBA4LB393P

6:29pm  Waiting for EC2 instances to launch. This may take a few minutes.

6:29pm  Created Auto Scaling group named:
        awseb-e-3ciwcwamzp-stack-AWSEBAutoScalingGroup-LKKPF6NPBICM

6:28pm  Created Auto Scaling launch configuration named:
        awseb-e-3ciwcwamzp-stack-AWSEBAutoScalingLaunchConfiguration-1AZRL03ZVD5NQ

6:28pm  Created security group named:
        awseb-e-3ciwcwamzp-stack-AWSEBSecurityGroup-F9KELKC4QULE

6:27pm  Environment health has transitioned to Pending. Initialization in progress (running for 4 seconds). There are no instances.

6:27pm  Created security group named:
        sg-0e539b2a158350f7c

6:27pm  Created target group named:
        arn:aws:elasticloadbalancing:us-east-1:618543464581:targetgroup/awseb-AWSEB-1SFEQ4Q54HYJ6/eb6a8ebb17fc97ef

6:27pm  Using elasticbeanstalk-us-east-1-618543464581 as Amazon S3 storage bucket for environment data.

6:27pm  createEnvironment is starting.
```

ONE ENVIRONMENT HAS SAMPLE APPLICATION DEPLOYED.

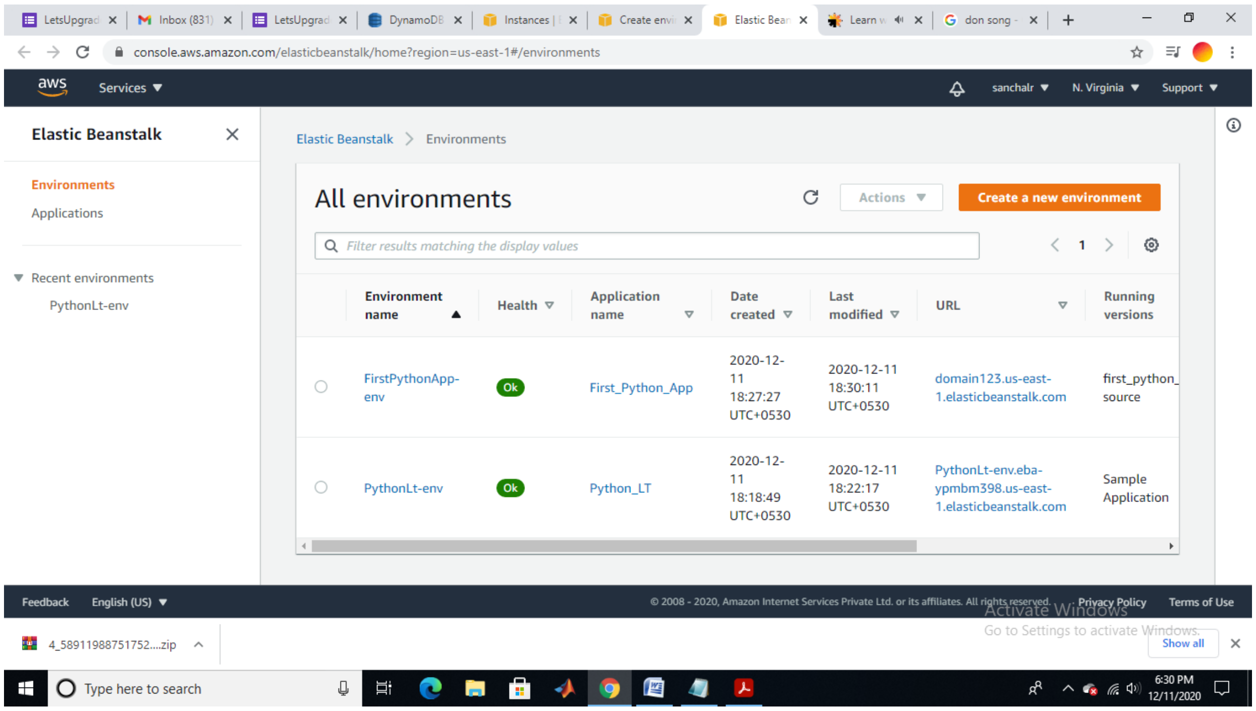


PythonLt-env
PythonLt-env.eba-ypmbm398.us-east-1.elasticbeanstalk.com (e-mwugqdness)
Application name: Python_LT

Health: Ok
Running version: Sample Application
Platform: Python 3.7 running on 64bit Amazon Linux 2/3.1.3

Recent events

BELOW SCREENSHOT SHOWS 2 ENVIRONMENTS ARE CREATED.



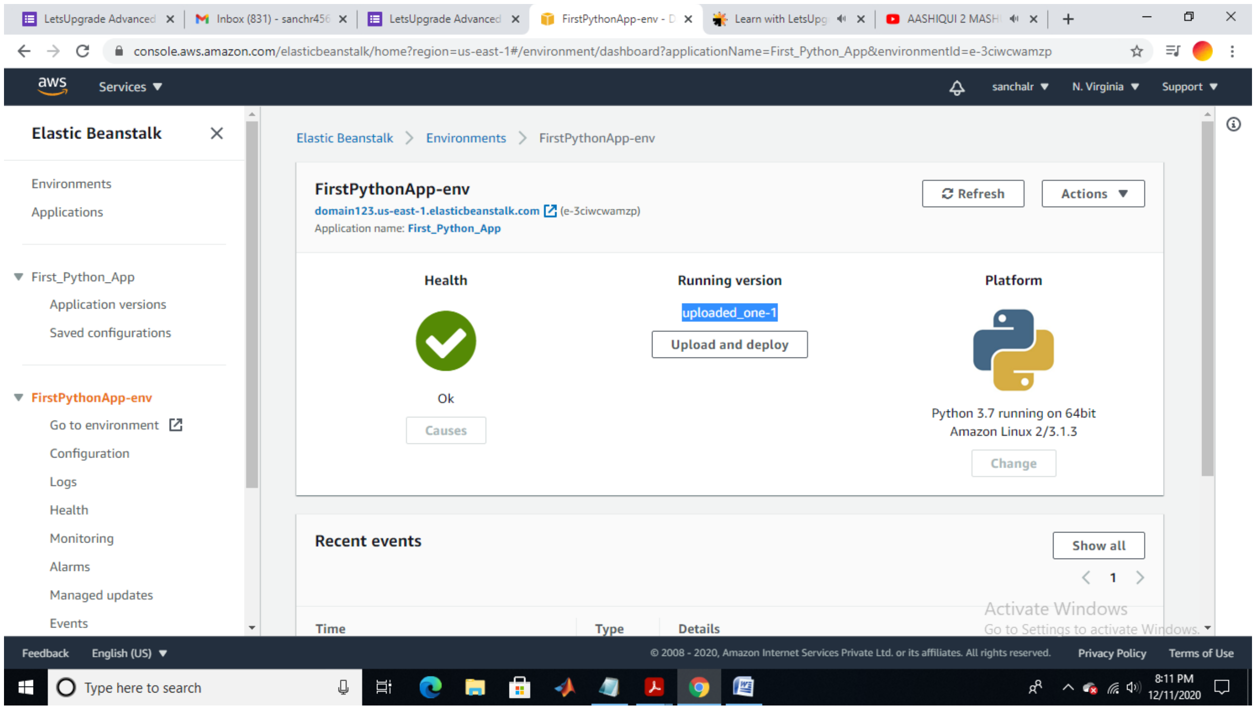
console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environments

Elastic Beanstalk > Environments

All environments

Environment name	Health	Application name	Date created	Last modified	URL	Running versions
FirstPythonApp-env	Ok	First_Python_App	2020-12-11 18:27:27 UTC+0530	2020-12-11 18:30:11 UTC+0530	domain123.us-east-1.elasticbeanstalk.com	first_python_source
PythonLt-env	Ok	Python_LT	2020-12-11 18:18:49 UTC+0530	2020-12-11 18:22:17 UTC+0530	PythonLt-env.eba-ypmbm398.us-east-1.elasticbeanstalk.com	Sample Application

HEALTH CHECK OF ENVIRONMENT IS REQUIRED AS THE OUTPUT OR ERRORS CAN BE RECTIFED AND SOLVED THEREAFTER.



console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/dashboard?applicationName=First_Python_App&environmentId=e-3ciwcwamzp

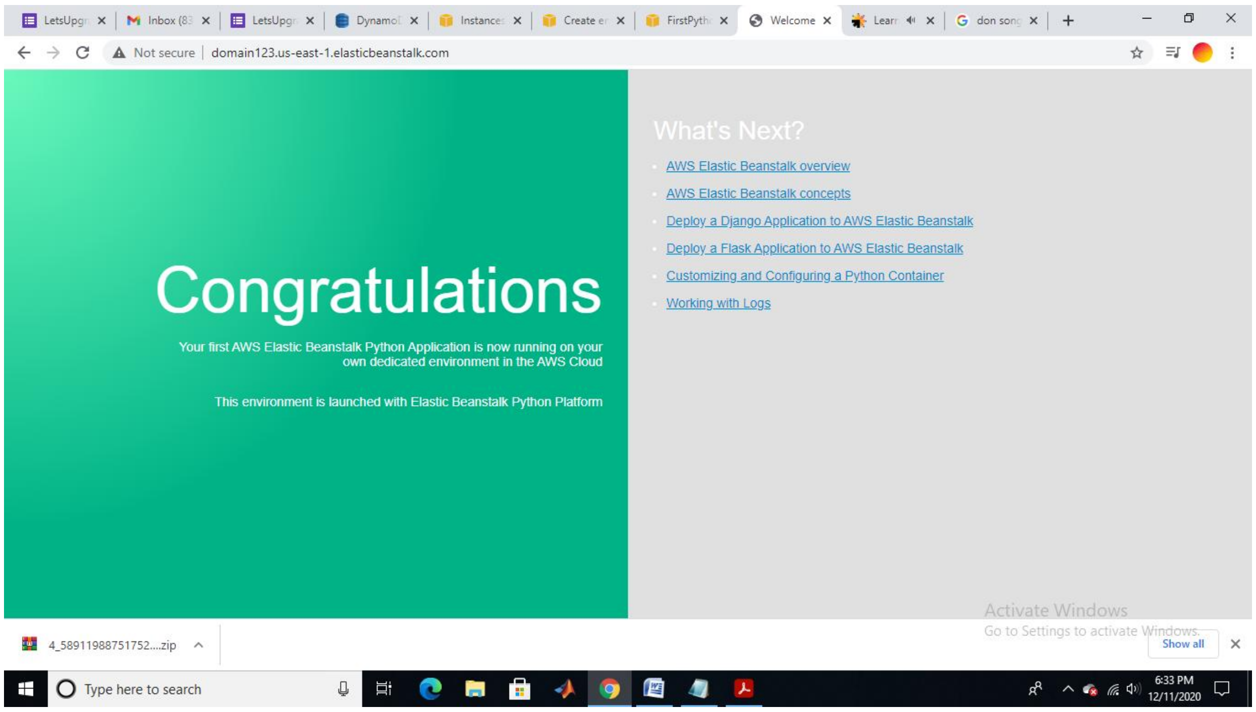
Elastic Beanstalk > Environments > FirstPythonApp-env

FirstPythonApp-env
domain123.us-east-1.elasticbeanstalk.com (e-3ciwcwamzp)
Application name: First_Python_App

Health: Ok
Running version: uploaded_one-1
Platform: Python 3.7 running on 64bit Amazon Linux 2/3.1.3

Recent events

<u>AFTER SUCCESSFUL UPLOAD AND DEPLOY, THE DEPLOYED APPLICATION IS DISPLAYED VIA DNS NAME OF ELASTIC BEANSTALK SELECTED ENVIRONMENT.</u>



Congratulations

Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Python Platform

What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy a Django Application to AWS Elastic Beanstalk
- Deploy a Flask Application to AWS Elastic Beanstalk
- Customizing and Configuring a Python Container
- Working with Logs