Documentation for IDGenerator

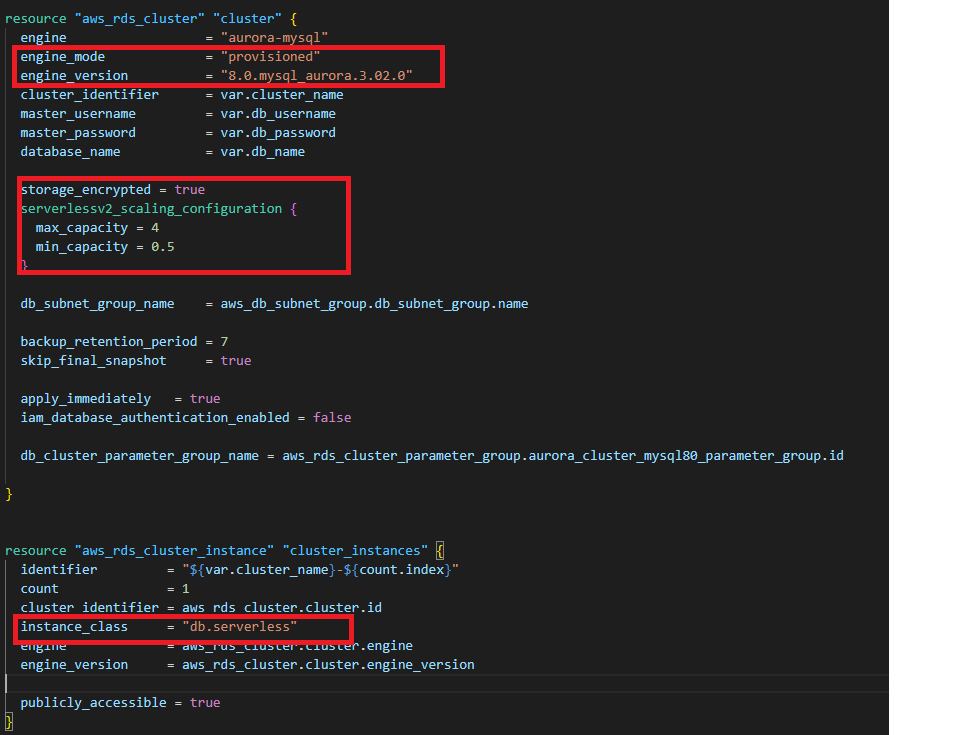
IDGenerator is the AWS Cloud based project which generates different Ids for applications like GuideWire, Surety, Arden and APS.

This document describes the various components of this project. This is followed by an explanation of how the main lambda function of this project works. Project deployment instructions follow and these include the RDS schema being created for this project.

# Serverless component

Aws Aurora serverless v2 is an on-demand and autoscaling configuration database. It is highly scalable, cost-efficient, and highly available in different zones. Aurora serverless can start and stop the computation automatically as per requirement and also allocate storage memory efficiently.

Terraform script for Aurora serverless is shown below.



# Database Component

## AWS RDS

AWS RDS is a fully managed AWS database service. It provides cost-efficient, scalable, and manageable relational database. We are using AWS RDS Aurora serverless V2, which is available for Aurora MySQL edition.

# General Component

## AWS VPC

Amazon’s virtual private cloud allows creation of resources for the virtual network and we have complete control over virtual environment for creating subnets, configuration, route tables, and network gateway etc.

## Aws Lambda

AWS Lambda is a serverless computing service, it is used to create lambda functions, self-contained applications that can be written in a language supported by AWS. The execution of the lambda function is efficient and flexible. The lambda function can perform computation tasks, serve web pages, and integrate with other AWS services.

# How Lambda Function works

Lambda function is using 9 different methods like idGenerate, updateReservedIds, updatePolicyIds, updateSequence, getData, getDataParams, getDataById, invoke\_lambda, and lambda\_handler.

## idGenerate

This method generates Id for applications like GuideWire, Surety, Arden and APS using Auto increment and saves that Id with current date, time and user information in a separate table for each application.

## updateReservedIds

This method is used to update reserve id in database. It takes 6 parameters namely, latestId, startId, endId, tbl\_name\_for\_sequence, appName and type. Here startId is the starting point of reserved Id and endId is the ending point of reserved Id and latestId is the currentId which will be updated in sequence table. Here this method is checking if latestId is between startId and endId, if so, it will update currentvalue in sequence table otherwise it will ignore latestId.

## updatePolicyIds

This method is used to update Policy Id in sequence table and also checks whether the latestId falls between the startId and endId values for Arden and Surety applications. If so, ignore the latestId and skip to the next number after endId and return. Otherwise (latestId not between startId and endId), simply increment by 1 and return the resulting number.

## updateSequence

This method gets latestId from idGenerate method and provides that Id to updateReservedIds method and updatePolicyIds method according to condition.

## changeSeedValues

This method changes the EndValue of Arden Account and Surety Account. Before changing the EndValue it checks whether this Id has already been used up by Guidewire policy values. If so, it will deny the request, otherwise it will update the EndValue of Arden Account and Surety Account.

## getData

This method is used to get current Id values of all applications from sequence table.

## getDataParams

This method is used to get last record for any Application like GuideWire, Arden and Surety from the RDS database.

## getDataById

This method is used to get record of specific Id for any application like GuideWire, Arden and Surety from the RDS database.

## invoke\_lambda

This method checks if sequence table exists in RDS or not. If sequence table exists in RDS it simply gets all data from the table otherwise it will create a table with the name of sequence in RDS and then insert a record in that table and then get all data from it and display in browser. It sends a response body to updateSequence method.

## lambda\_handler

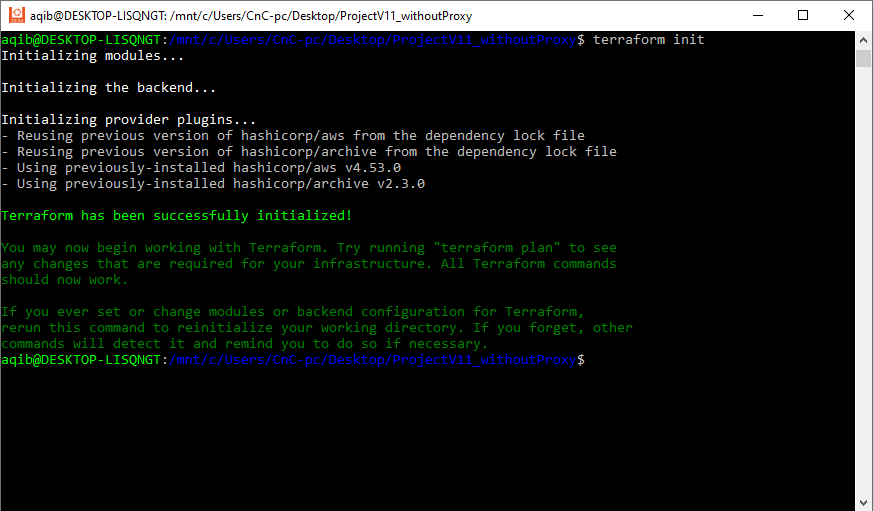
This is the main function of lambda function. It has all information of API request body and request details that is used to invoke lambda.

# How to Deploy the project

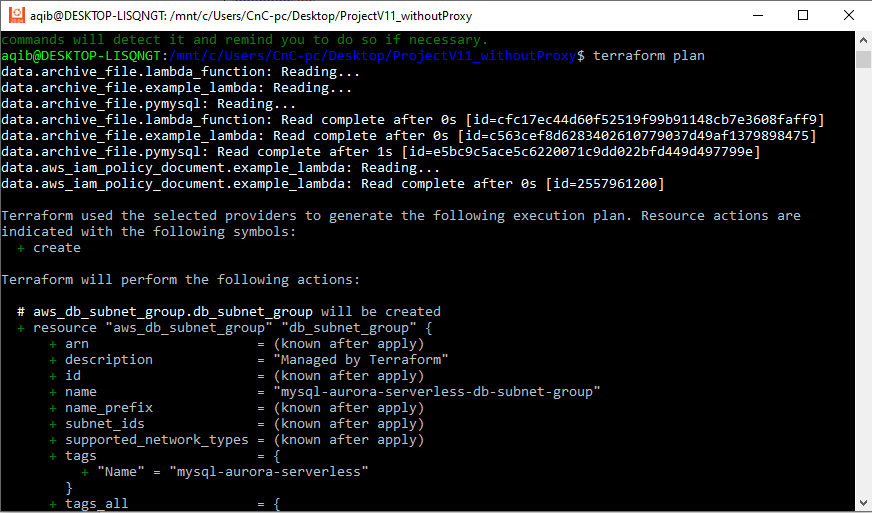
This Project is deployed by Terraform. Terraform is a tool for building, changing, and managing infrastructure in a safe, repeatable way. It is an infrastructure provisioning tool where you can store your cloud infrastructure setup as codes.

The following commands are used to deploy the project:

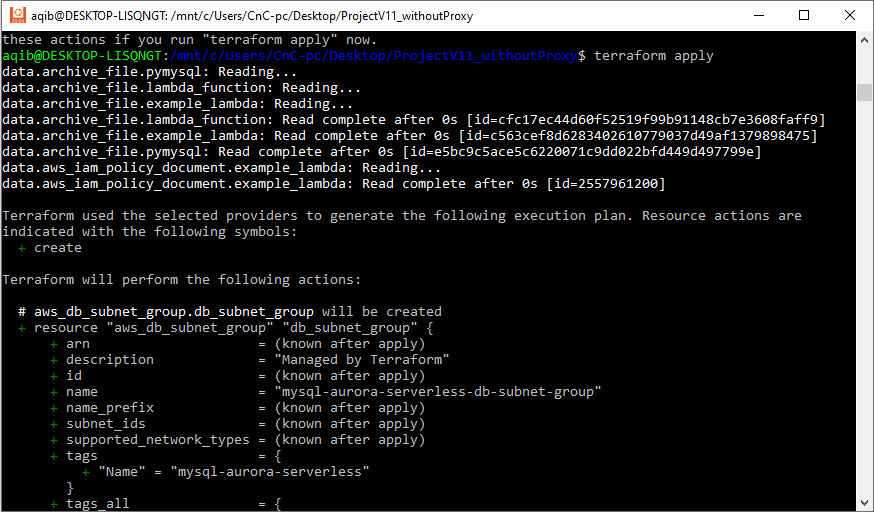
**$ terraform init**

****

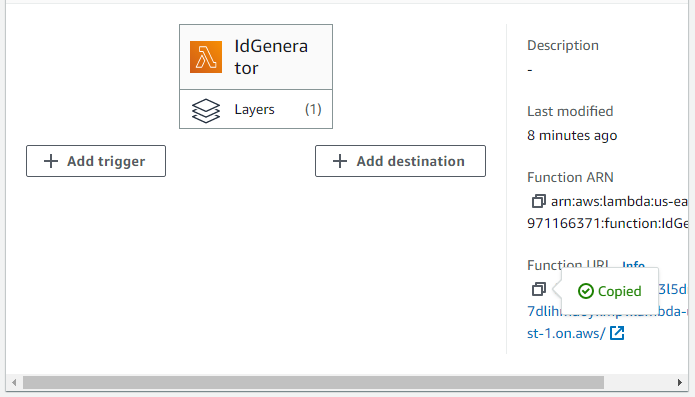
**$ terraform plan**

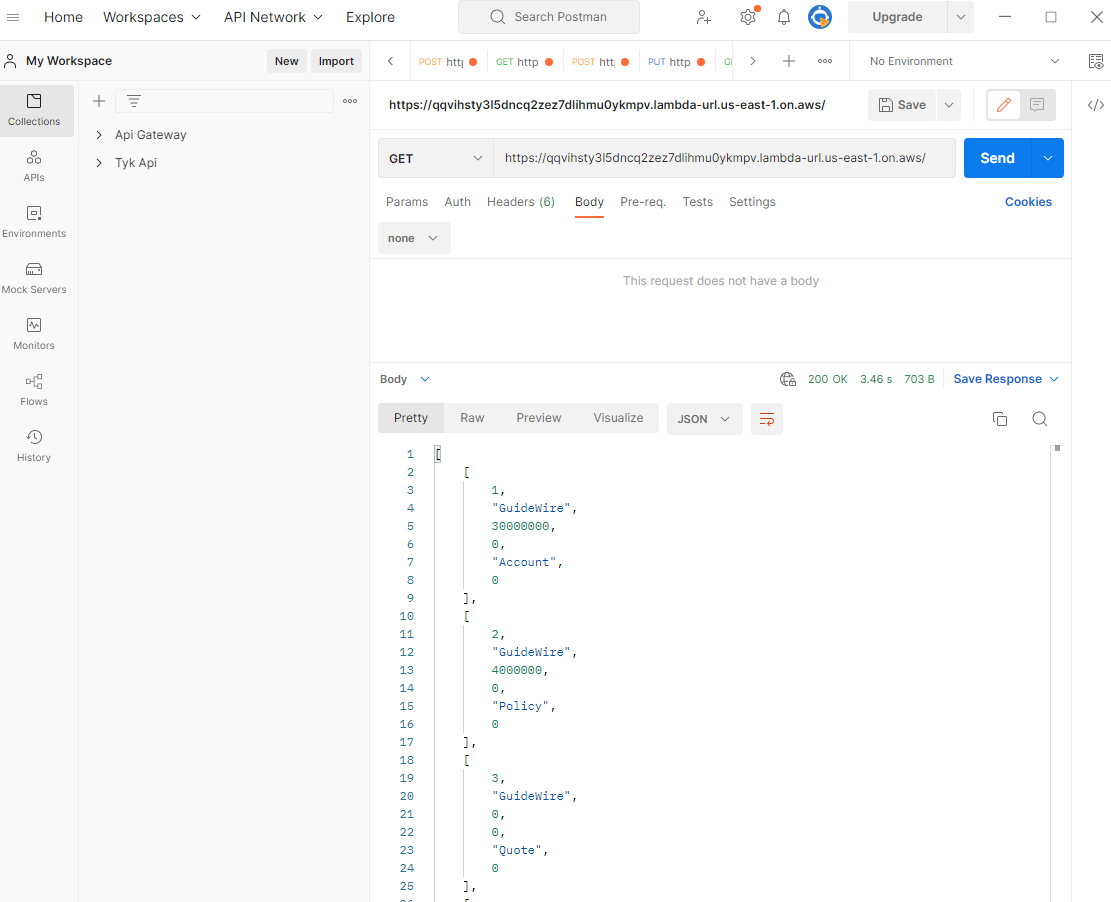


**$ terraform apply**



After "**terraform apply**" command completes, copy the Lambda function **url** and use it to call **GET** method through browser or Postman.

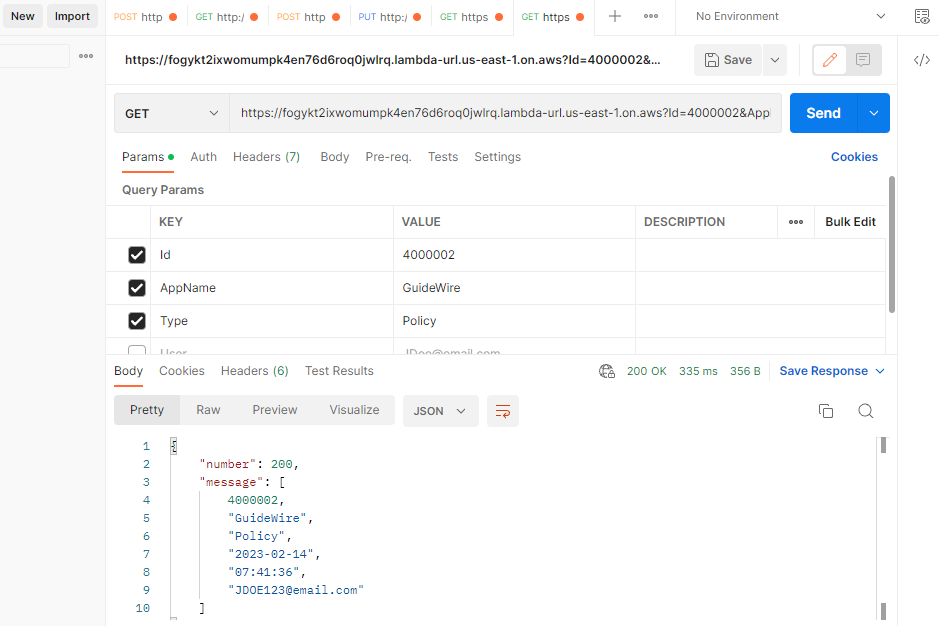




To get data from database based on the **Id, s**end a parametrized **GET** request. Url of the request looks like below:

**https://fogykt2ixwomumpk4en76d6roq0jwlrq.lambda-url.us-east-1.on.aws?Id=4000002&AppName=GuideWire&Type=Policy**

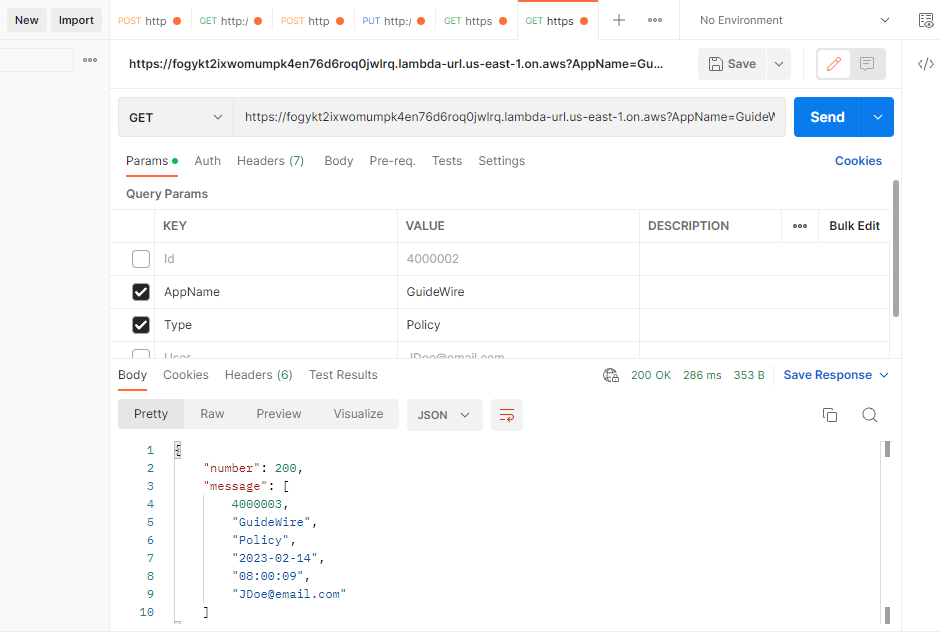
It returns detail of the Id for the AppName and Type provided (if it exists)



To get last record of any application from the database, send a parametrized **GET** request. Url of the request looks like below:

**https://fogykt2ixwomumpk4en76d6roq0jwlrq.lambda-url.us-east-1.on.aws?AppName=GuideWire&Type=Policy**

It returns audit detail of last Id generated for the provided AppName and Type.



In the **POST** method of Lambda function, send parameters in JSON format.

Details of REST API parameters:

**AppName**: This is the application name e.g. **GuideWire**, **Surety**, **Arden** and **APS**.

**Type**: This is Id Type e.g. **Account**, **Policy** and **Quote**.

**User**: This contains user details e.g. email address or any other system info

* Send POST request with the following parameters:

{

“User”:”JDOE123@email.com”,

“AppName”: “GuideWire”,

“Type”: “Account”

}

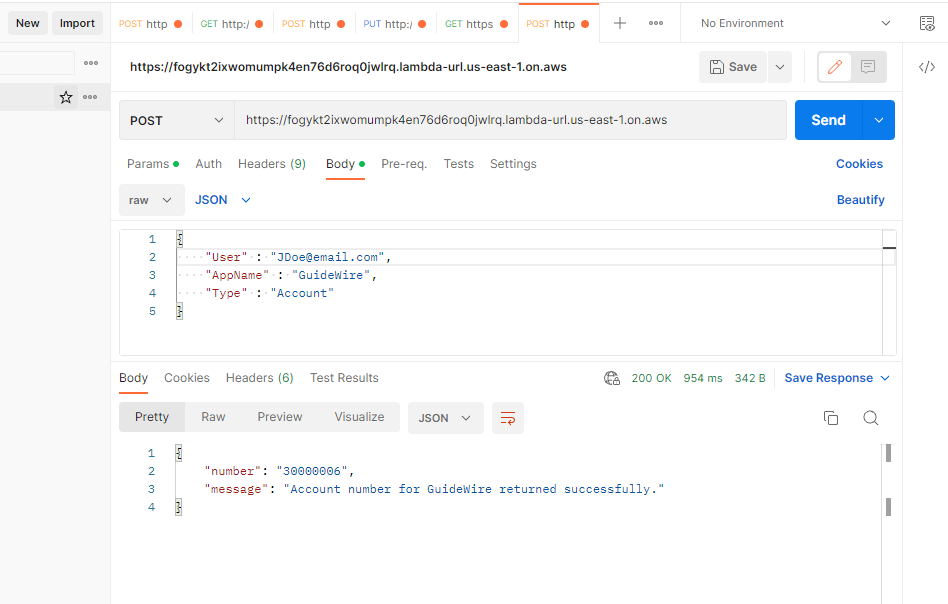
* It request returns data in json format as shown below:

{

“number” : ” 30000006”,

“message”: “Account number for GuideWire returned successfully.”

}



Update the End value of Arden Account and Surety Account as follow:

* Send PUT request with the following parameters:

{

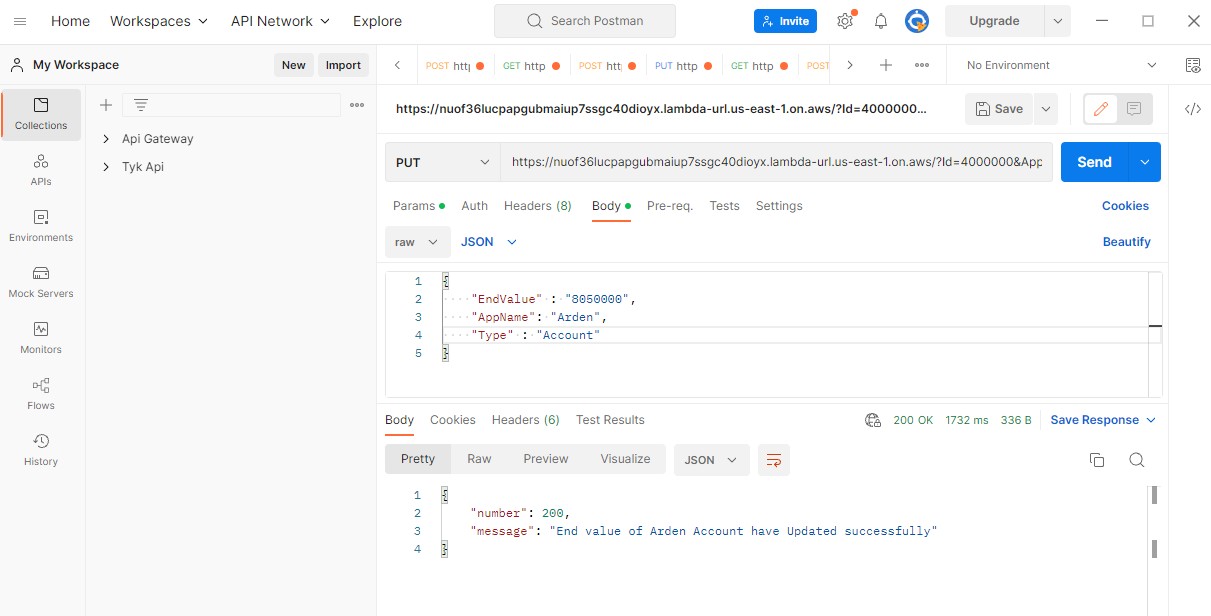
"EndValue": "8050000",

"AppName": "Arden",

"Type": "Account"

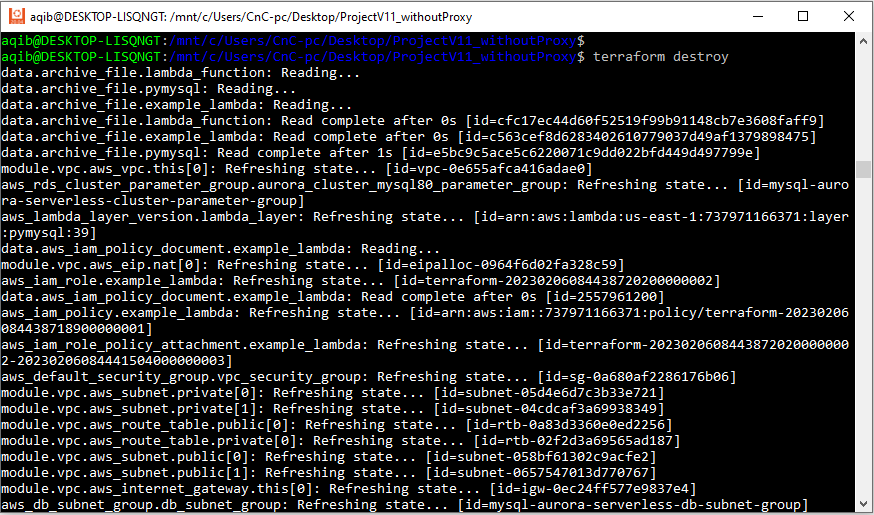
}

* The above request updates the End Value of Arden Account.



In case complete project is to be deleted, run the following command:

**$ terraform destroy**



**Schema for RDS**

Lambda function creates following 5 tables in the RDS database:

1. GuideWireAccountIDS
2. GuideWirePolicyIDS
3. SuretyAccountIDS
4. ArdenAccountIDS
5. Sequence

### **Table for sequence**

Columns in the sequence table are (Id, AppName, Type, StartId, EndId and CurrentValue) and it is used to store start, end and the latest Id value of GuideWireAccountIDS, GuideWirePolicyIDS, SuretyAccountIDS and ArdenAccountIDS tables.

### **Table for GuideWireAccountIDS**

Columns in the GuideWireAccountIDS table are (Id, AppName, Type, Date, Time and User) and it is used to generate a new Id and save application name, type of id, current date, time and user information in the GuideWireAccountIDS table.

### **Table for GuideWirePolicyIDS**

Columns in the GuideWirePolicyIDS table are (Id, AppName, Type, Date, Time and User) and it is used to generate a new Id and save application name, type of id, current date, time and user information in the GuideWirePolicyIDS table.

### **Table for SuretyAccountIDS**

Columns in the SuretyAccountIDS table are (Id, AppName, Type, Date, Time and User) and it is used to generate a new Id and save application name, type of id, current date, time and user information in the SuretyAccountIDS table.

### **Table for ArdenAccountIDS**

Columns in the ArdenAccountIDS table are (Id, AppName, Type, Date, Time and User) and it is used to generate a new Id and save application name, type of id, current date, time and user information in the ArdenAccountIDS table.