# TLC303 tutorial

Welcome to TLC303 !

before we get started , make sure your wifi is working properly .

you would need to work in teams of 3 , find yourself two people who are sitting next to you , those will be your partners for this session. in order to build the data mesh architecture which we will then use as a foundation for telcolake use cases , we need to have 3 AWS accounts , which we will provide. based on your seating order , the leftmost person will be the Producer account , the one sitting in the middle will be Central data mesh account , and the person on the right will be the consumer account.

Producer→Data Mesh(Central) → Consumer

after we finish the part of building a data mesh architecture , most of the focus will be on building 3 different telco use cases , leveraging the different datasets , hence you can then work together on the consumer account as a team .

The AWS Data Mesh Helper library provides automation around the most common tasks that customers need to perform to implement a data mesh architecture on AWS. A data mesh on AWS uses a central AWS Account (the mesh account) to store the metadata associated with **Data Products** created by data **Producers**. This allows other AWS Accounts to act as  **Consumers**, and to request **Subscriptions**, which must be approved by **Producers**. Upon approval, the approved grants are provided to the **Consumer** and can be used within their AWS Account.

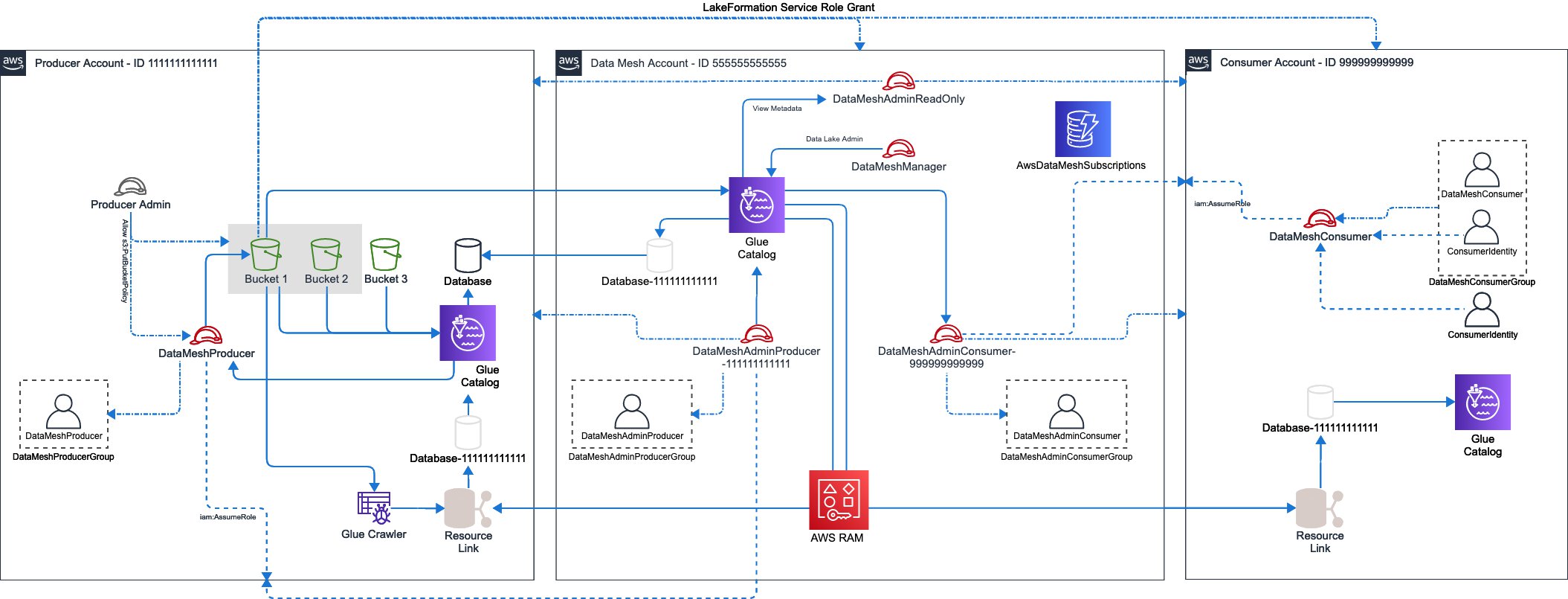
### Definition of Terms

* **Data Mesh** - An architectural pattern which provides a centralized environment in which the data sharing contract is managed. Data stays within **Producer** AWS Accounts, and they own the lifecycle of granting **Subscriptions**.
* **Producer** - Any entity which offers a **Data Product** through the **Data Mesh**
* **Consumer** - Any entity who subscribes to a **Data Product** in the **Data Mesh**
* **Subscription** - The central record and associated AWS Lake Formation permissions linking a **Data Product** to a  **Consumer**
* **Data Product** - Today, a **Data Product** is scoped to be only an AWS Lake Formation Table or Database. In future this definition may expand.

### The Workflow

To get started, you must first enable an AWS Account as the **Data Mesh** Account. This is where you will store all Lake Formation metadata about the **Data Products** which are offered to **Consumers**. Within this Account, there exist IAM Roles for **Producer** and **Consumer** which allow any AWS Identity who has access to perform tasks within the Data Mesh.

Once you have setup an Account as the **Data Mesh**, you can then activate another AWS Account as a **Producer**,  **Consumer**, or both. All of these tasks are performed by the **Data Mesh Admin**, which is accessible through an additional IAM Role or as any Administrator Identity within the mesh Account. Once completed, end users can perform the following Data Mesh tasks:



In this architecture, we can see that the data mesh is configured in AWS Account 555555555555, and contains a set of IAM Roles which allow identities within producer and consumer accounts to access the mesh. This includes:

* DataMeshManager: IAM Role allowing administration of the Data Mesh itself
* DataMeshAdminProducer: IAM Role enabling the assuming Identity to act as a **Producer**
* DataMeshAdminConsumer: IAM Role enabling the assuming Identity to act as a **Consumer**
* DataMeshAdminReadOnly: IAM Role that can be used for reading Metadata from the Data Mesh Account (only)

For testing and simplicity, every IAM Role in the solution is accompanied by a single IAM User who is a member of a Group specific to the function. This will enable you to add users to this Group should you wish to, rather than using a programmatic approach. IAM Roles are backed by an IAM Policy of the same name as the Role, and all objects in the IAM stack for AWS Data Mesh reside at path */AwsDataMesh/*.

You can then see that there is a Producer Account 111111111111 who has been enabled to act as a **Producer**. Within this account we see a similar approach to IAM principals, with the creation of a DataMeshProducer IAM Role which is accompanied by an associated user and group. When configured, the DataMeshProducer group is granted rights to assume the DataMeshProducer-<account id> role in the data mesh Account.

Similarly, we have a consumer Account 999999999999. This Account also includes IAM objects to enable data mesh access, including the DataMeshConsumer IAM Role, and associated IAM users and groups. Only the DataMeshConsumer role may assume the DataMeshAdminConsumer-<account id> role in the data mesh Account.

All information around current or pending subscriptions is stored in DynamoDB, in table AwsDataMeshSubscriptions. This table is secured for only those operations which Producers or Consumer roles are allowed to execute, and stores the overall lifecycle for Subscriptions.

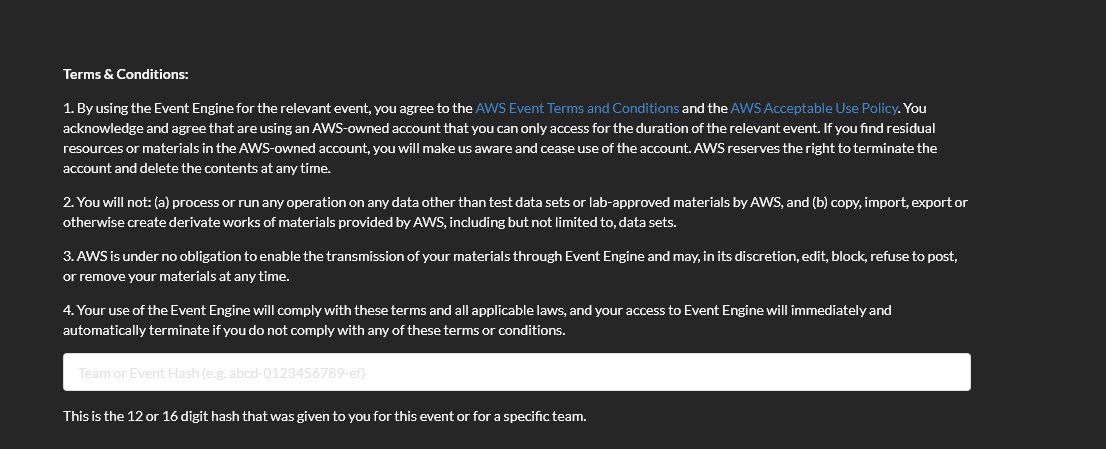
under the home directory  \Data-mesh-util\test\reinvent you will find various tests that show the functionality of the Data Mesh Utility, including:

* data\_mesh\_account\_tests.py: Tests which set up the central Data Mesh Account
* producer\_account\_tests.py: Tests for the Producer side of data production creation and subscription access
* consumer\_account\_tests.py: Tests for a Consumer to request access to a product and import subscription objects
* integration.py: Full integration test showing configuration of the central mesh account, creation of a data product, request of access, and granting access for a consumer.

## Getting Started

## Getting started with Event Engine

open your web browser and enter the following URL : <https://dashboard.eventengine.run/login>



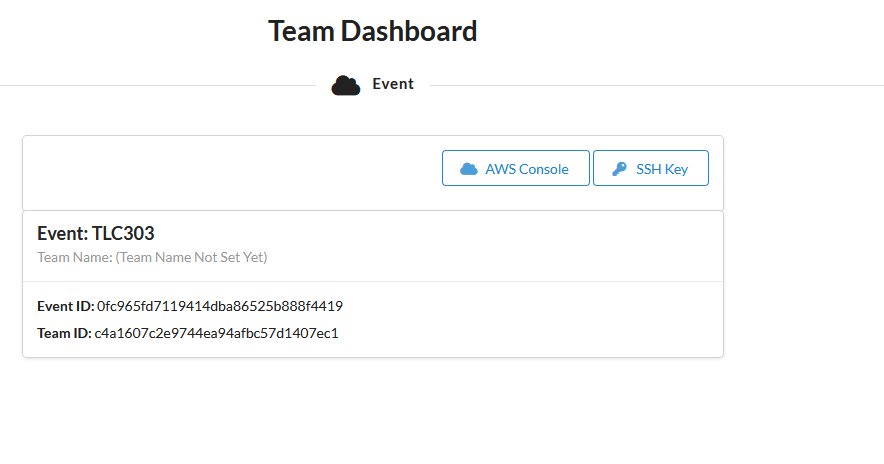
please enter the Event Hash that will be provided .

Choose “Email One-Time Password (OTP")"

enter your email address

you should receive an email with your one time password , copy the password and paste it in the right location .

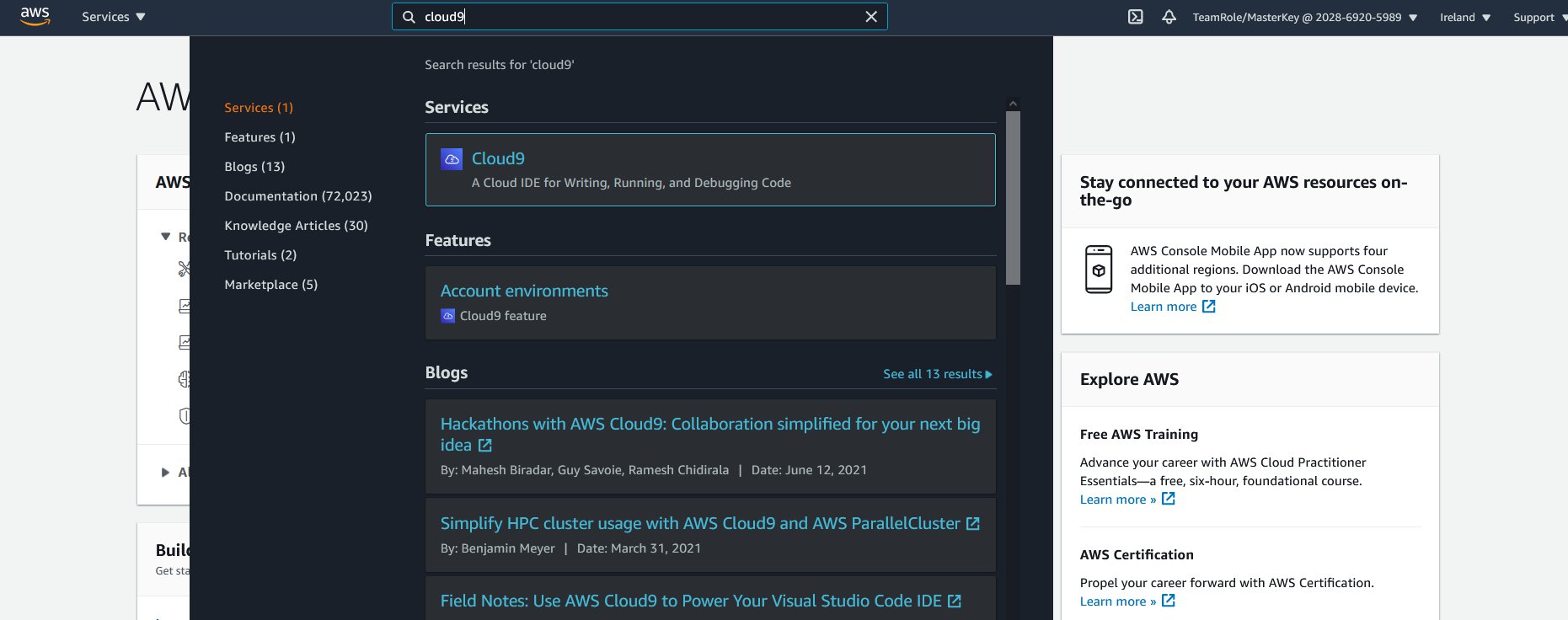
once done you should see the following page :



click on “AWS Console” and then on “Open AWS console”. that should open up the AWS console , and you should be good to start the workshop !

first we will need to start by setting up an IDE which will be used for us to run the datamesh scripts . for that we will use AWS cloud9.

from the search bar , search for cloud9 and click on it .



click on **Create Environment**

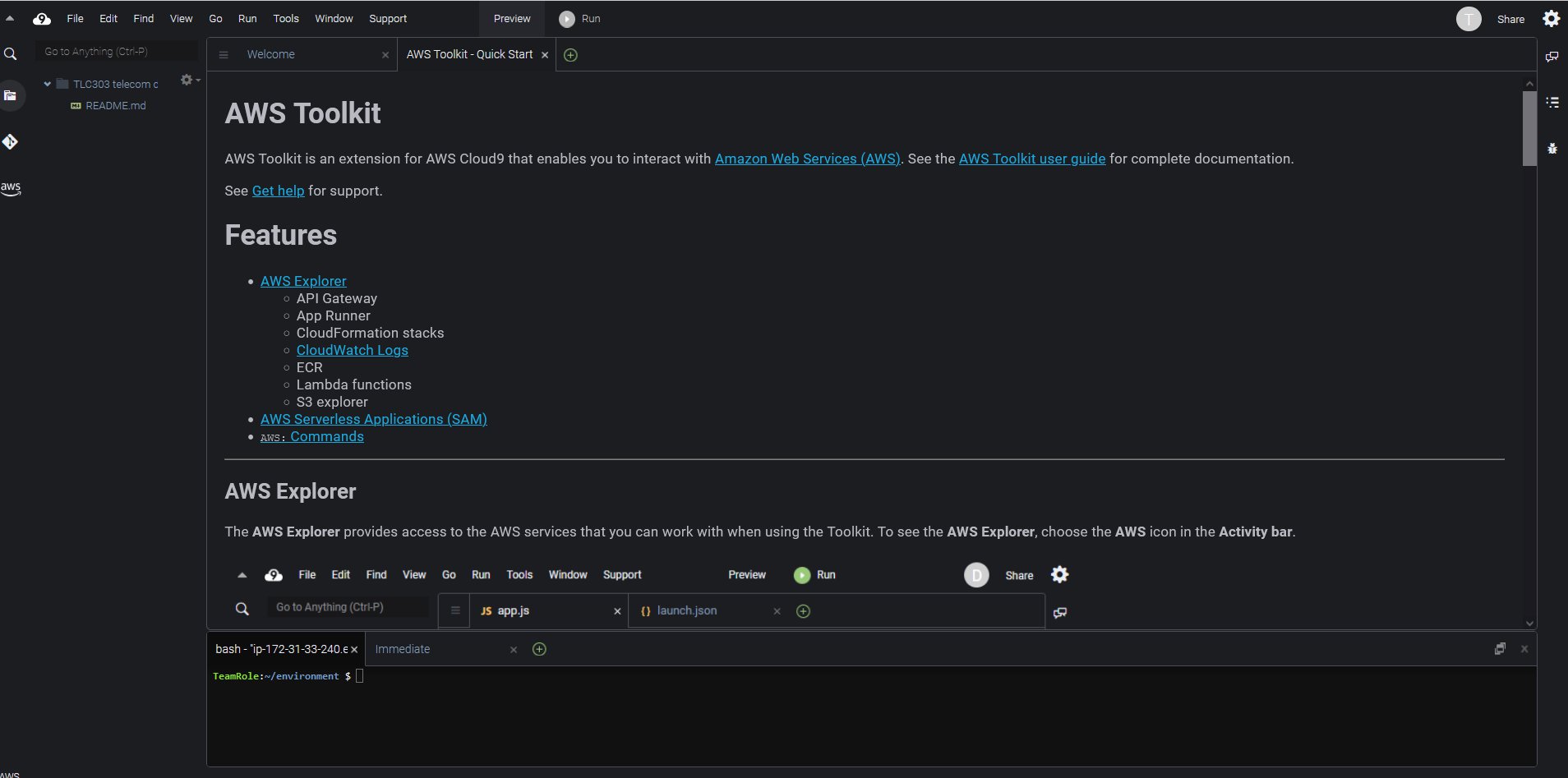
Choose a name for the environment and click **Next Step**

change the instance type to **t3.small**

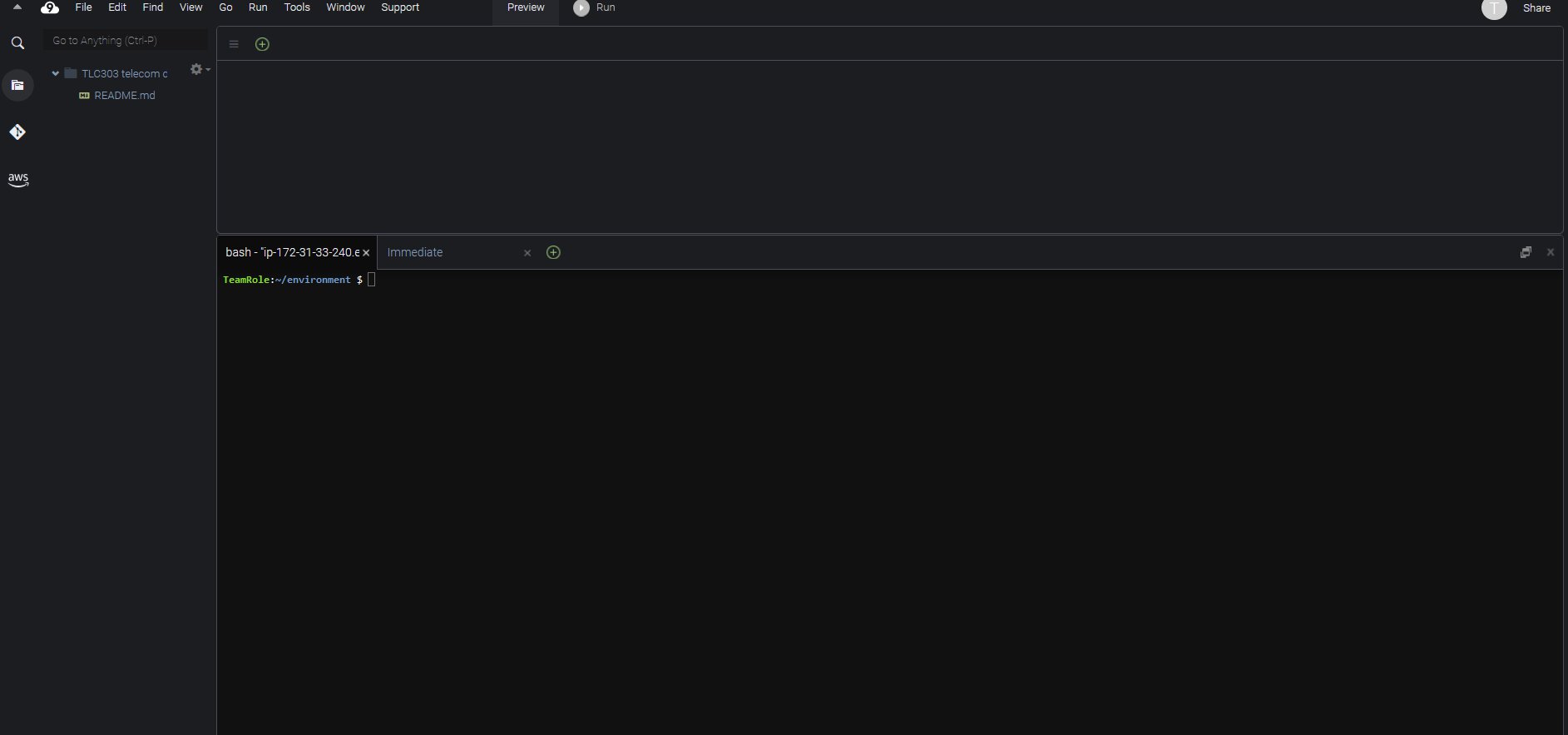
click on **Next Step**

verify the details are correct and click on **Create Environment**

the cloud9 will take a couple of minutes to load , once done you should see the following screen :



you can close the 2 tabs on the top of the page and you can grab the terminal tab on the bottom and extend it :



now , we would like to copy the source code  -**use Git or S3**

go to the terminal , make sure you are in the data-mesh util directory . and the following command :

pip3 install -r requirements.txt

that will install the relevant packages required to run the data mesh util.

Each step requires the configuration of a CredentialsFile environment variable, which is a JSON document on your filesystem that provides access to the Accounts to be used. The structure of this file is shown below (you must remove comments), or you can start with [sample\_test\_creds.json](https://code.amazon.com/packages/Data-mesh-util/blobs/mainline/--/test/sample_test_creds.json).

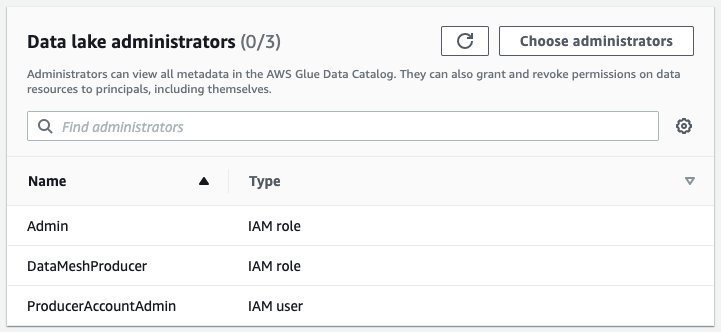
{  
  "AWS\_REGION": "us-east-1",  
  # must be LF data lake admin in the mesh account  
  "Mesh": {  
    "AccountId": "",  
    "AccessKeyId": "",  
    "SecretAccessKey": ""  
  },  
  "Producer": {  
    "AccountId": "",  
    "AccessKeyId": "",  
    "SecretAccessKey": ""  
  },  
  # must be LF data lake admin in a producer account  
  "ProducerAdmin":{  
    "AccountId": "",  
    "AccessKeyId": "",  
    "SecretAccessKey": ""  
  },  
  "Consumer": {  
    "AccountId": "",  
    "AccessKeyId": "",  
    "SecretAccessKey": ""  
  },  
  # must be LF data lake admin in a consumer account  
  "ConsumerAdmin": {  
    "AccountId": "",  
    "AccessKeyId": "",  
    "SecretAccessKey": ""  
  }  
}

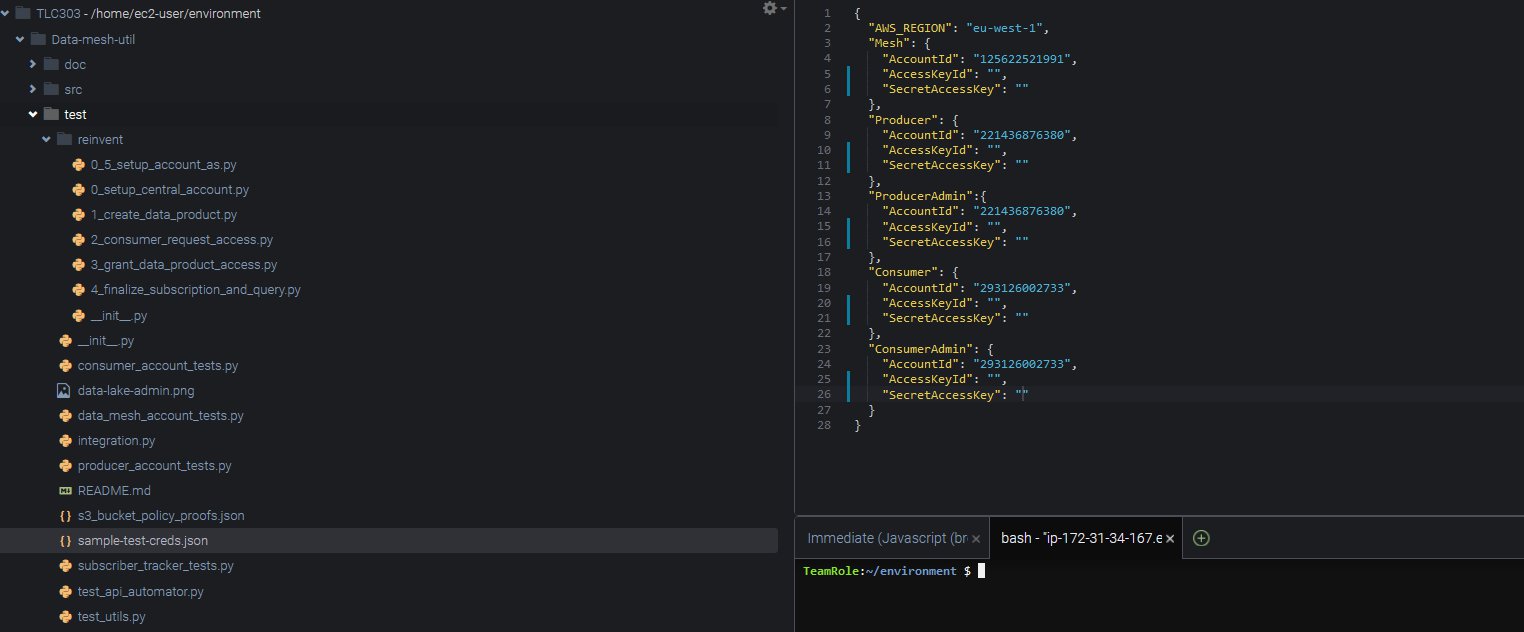
This file allows for the configuration of multiple accounts, for the Data Mesh, and then a "normal" user and Administration user for both the producer and consumer. Please note that the keys of this document are reserved and cannot be changed or extended (Mesh, Producer, ProducerAdmin, Consumer, and ConsumerAdmin are all reserved words).

In general, you should start by configuring administrative users Mesh, ProducerAdmin, and ConsumerAdmin administrators. You can then setup the core Data Mesh functionality, and then add Producer and Consumer entries from the respective accounts after enabling Access Keys for the created DataMeshProducer and DataMeshConsumer sample users.

Please note that Mesh, ProducerAdmin, and ConsumerAdmin must all be assigned Data Lake Admin permissions in Lake Formation. This permissions Grant falls out of the scope of this utility as it requires root or Data Lake Admin to assign. You can assign these permissions using the AWS Console in the Account you wish to configure.

In the AWS Console, choose the Lake Formation service, and then in the left-hand Nav choose Permissions/Administrative Roles & Tasks. Add the User or Role configured above as data lake administrator:





each user in his AWS account , will create an IAM user and will provide the relevant details , which will be populated in the datamesh account.

log in to your AWS console

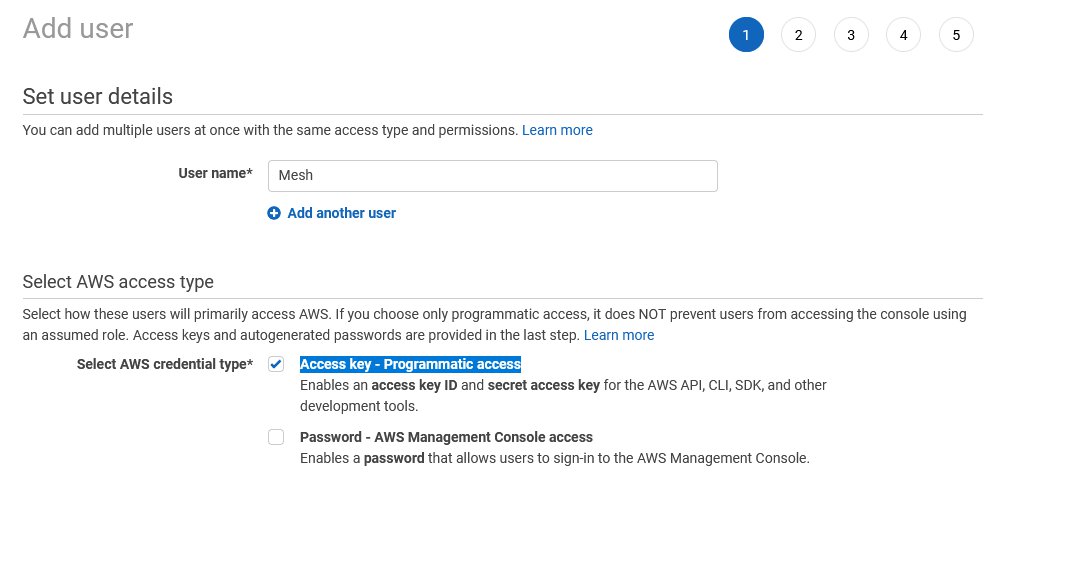
search for IAM , in the search bar and select IAM.

we will now create the users which are needed for the data mesh util.

select **Users** and then **add users**

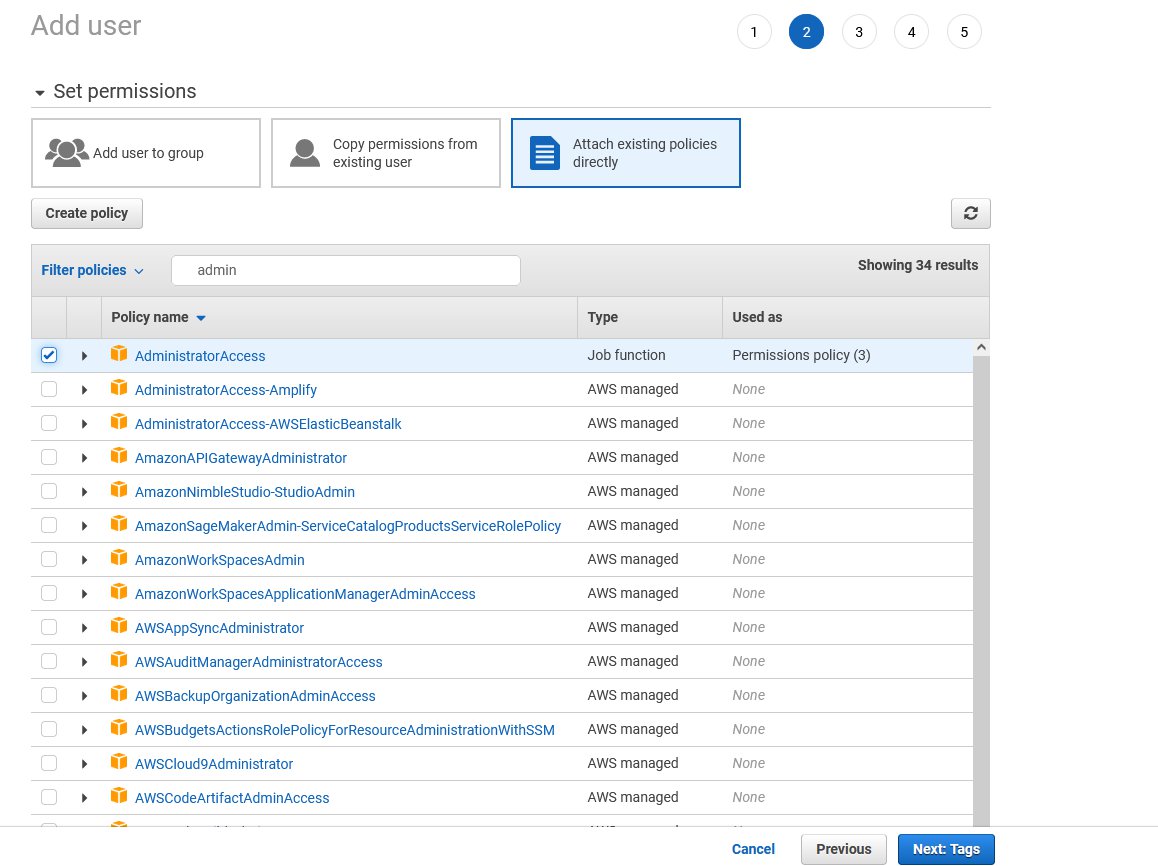
enter a user name , and try to make it identifiable - Producer for the producer account , consumer for the consumer account ,central/mesh for central account

under select AWS access type , select **Access key - Programmatic access** and click **next:permissions**



under Set permissions , click on **Attach existing policies directly** and checkbox **AdministratorAccess**

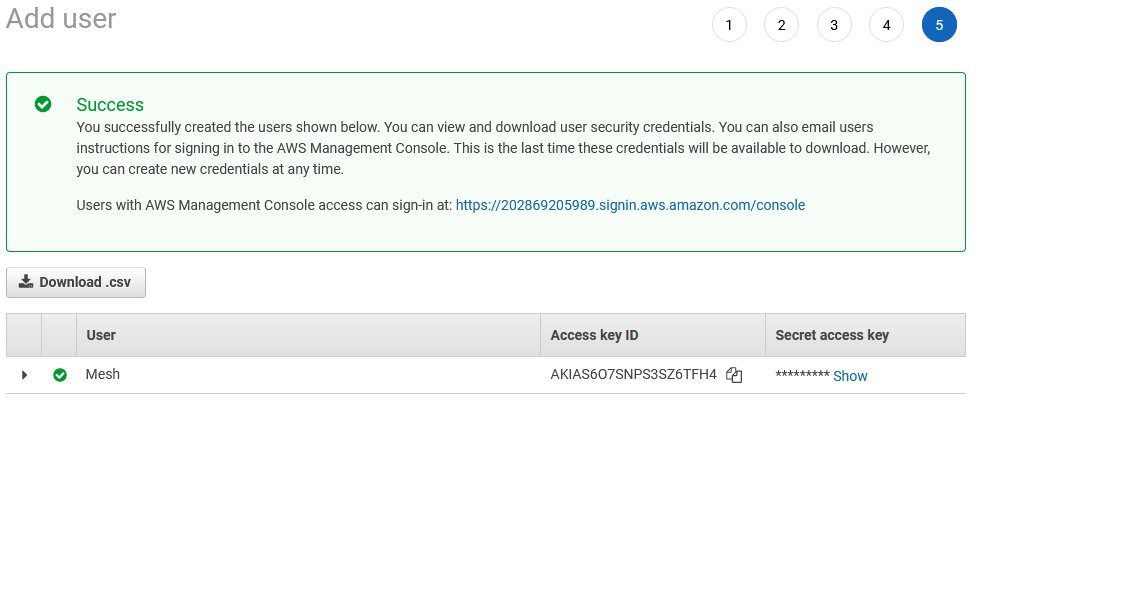
click on **Next:Tags**



click **Next** and in the last step **Create User**

once the user has been created, you will see a page showing the Access key ID and the secret access key .

please save those in a notepad , and share those with the mesh account owner so he can then populate the credentials file

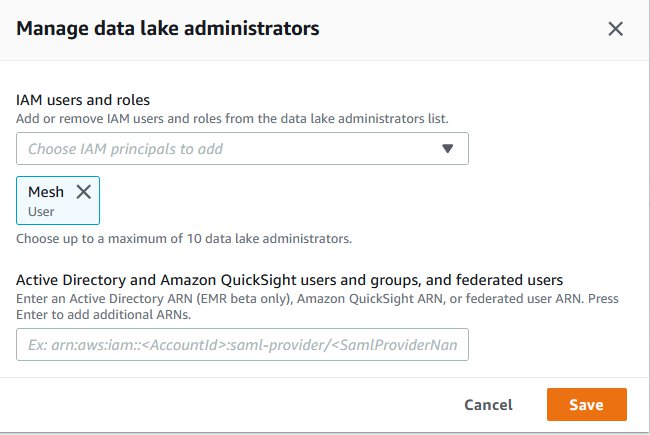


now go to the search bar and search for **Lakeformation** .

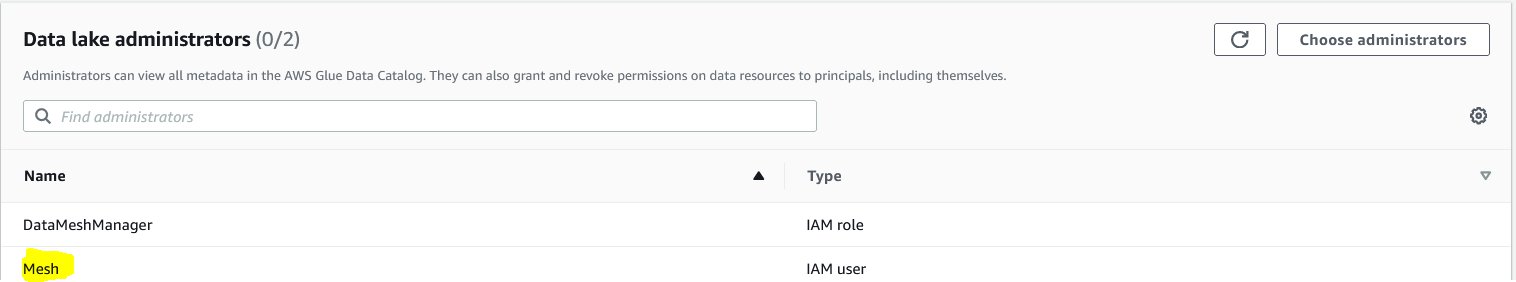
we will now provide datalake admin rights to the user we have just created .

click on the left side under Premissions on **Administrative roles and tasks** then click on **Choose administrators.**

select the relevant user which you have created and click **Save**



once done you should see that the IAM user now is a Data lake administrator:



on the mesh account , in cloud9 , open the **/Data-mesh-util/test/sample-test-creds.json** file

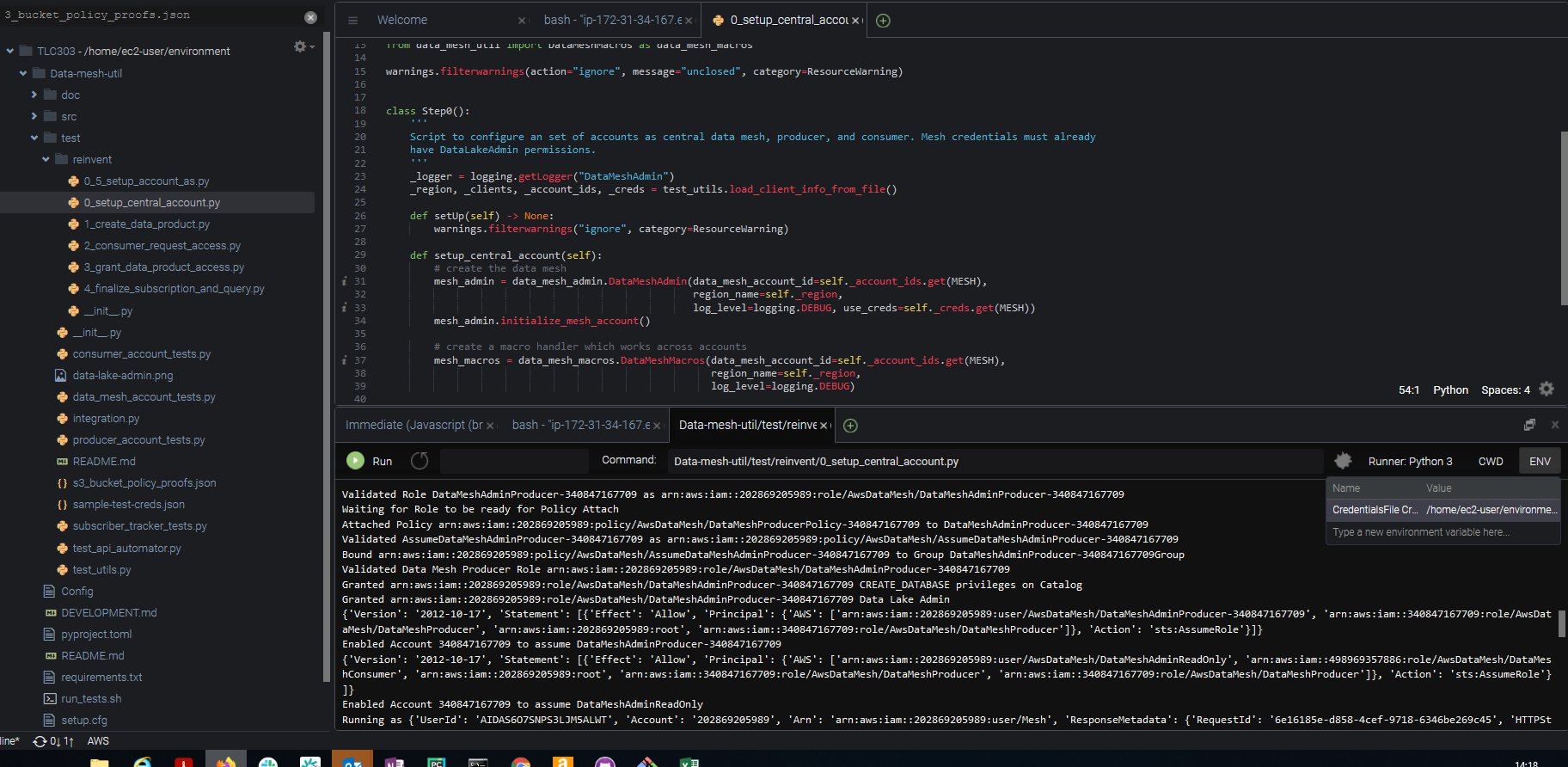
the Mesh, Producer Admin and Consumer Admin are the ones which should be populated .

once those are populated . we can start running the steps .

the first step is 0\_setup\_central\_account.py , click on the file and on the right side of the screen select **ENV.**

this is where we create an enviorment file which will point to our ceredntials file. set Name for **CredentialsFile** and Value to **/home/ec2-user/environment/Data-mesh-util/test/sample-test-creds.json**

once done click on **Run** and script will start running and executing the relevant tasks



once the first step was completed , you should see in the output ,the last line as :

Enabled Account xxxxxxxxxxxxx assume DataMeshAdminReadOnly

xxxxxxxxxxx should be your AWS consumer account ID.

go to your AWS producer/consumer accounts .

go to IAM , you should see a DataMeshProducer user  on the producer account and a DataMeshConsumer user in the consumer account.

click on the on the **user name**

select the **security credentials** tab

click on the **create access keys** button

make sure you copy the **Access key ID** and the **Secret access key**

the mesh account owner can now populate the producer and consumer account and credentials details .

once done you should now have the credentials file fully populated .

next step is to download the 3 datasets that we will be using for our telecom use cases.

we would need to :

go to the AWS producer account

download the datasets

run the AWS Glue crawler to discover the schemas and build a glue catalog:

in the search bar type glue and select **AWS Glue**

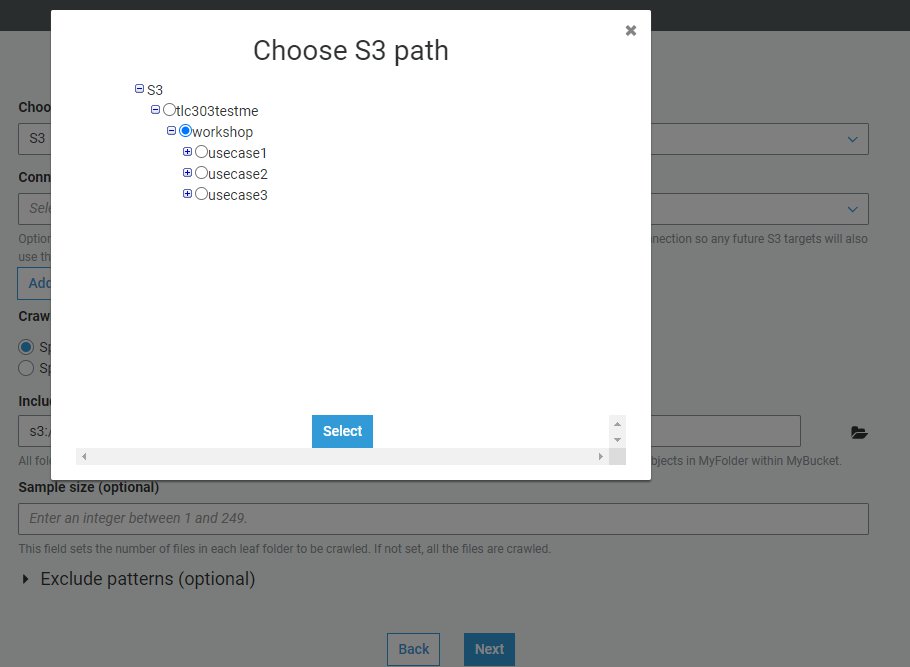
click on **Crawlers** under **Data catalog**

click on **Add crawler**

enter a **Crawler name** and click **next**

click **next** again

select the s3 path where the datasets for the use cases are located under **include path**



click **Next**

click **Next**

select **Create an IAM role**

enter a name for your Crawler IAM role and click **Next**

click **Next**

click on **Add database**

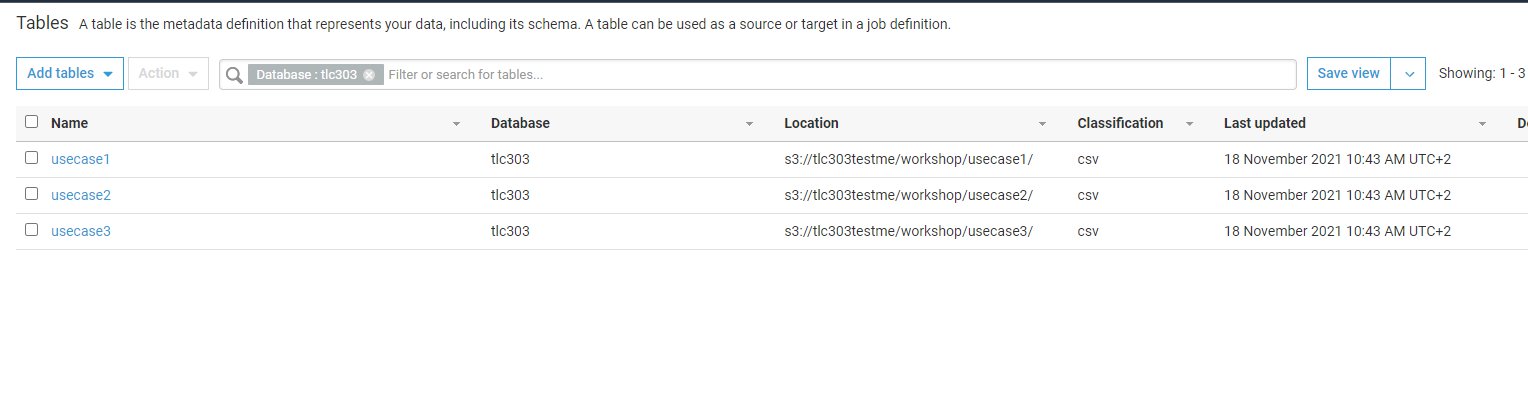
enter a **database name** and click **Create**

click **Next** and then **Finish**

your AWS Glue crawler is now ready , click on the checkbox next to the crawler name and then click on **Run Crawler**

the Glue crawler will run for approximately one mintue , and will discover the 3 datasets and create the tables in the glue catalog.

if you go now to to **Tables** under **Databases** and the **Data  Catalog ,** you will see 3 tables that were created by the crawler you can click on each one of them and explore the schema



once that is done , we have products ready on the producer side that can be shared with the central/Hash account

go to the mash account .

on the mesh account.

we would now prepare to run step 1\_create\_data\_product.py ,  where we will copy the glue catalog to the central mesh account from the producer account.

we first set the enviorment variable same as we did in step 0 ,double clickiing on the 1\_create\_data\_product.py and then clicking on **ENV**

this is where we create an enviorment file which will point to our ceredntials file. set Name for **CredentialsFile** and Value to **/home/ec2-user/environment/Data-mesh-util/test/sample-test-creds.json**

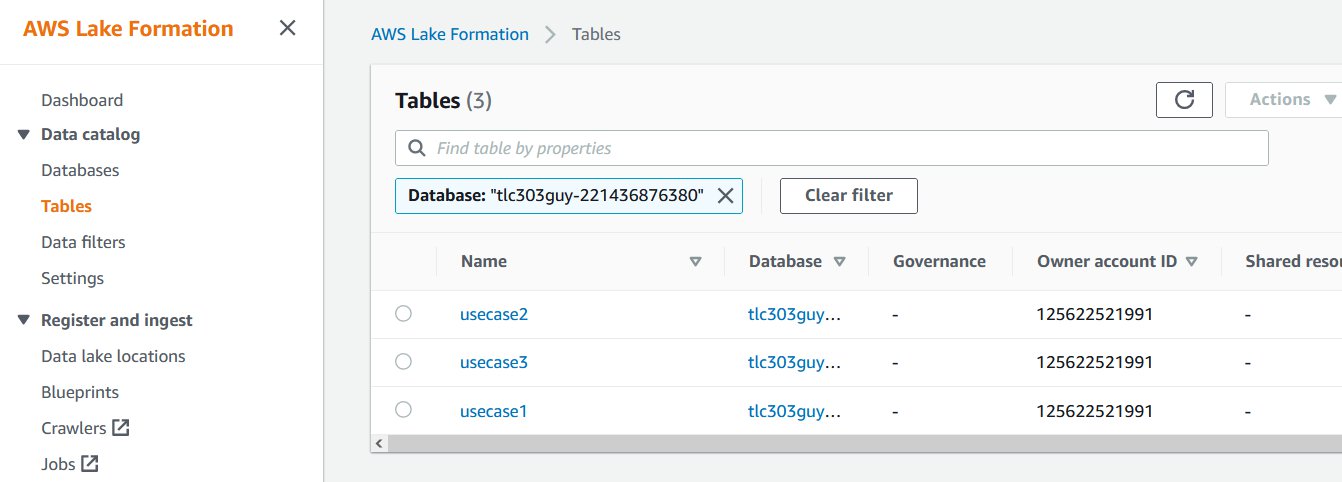
now , we need to tell our script what is the glue database\_name  and what is the table\_regex

you can add those parameters to the Command line similarly to the below :

Data-mesh-util/test/reinvent/1\_create\_data\_product.py --database\_name tlc303 --table\_regex usecase\*

once done click on **Run** and script will start running and executing the relevant tasks

once done , you can go to the mesh account , and validate you a database and tables were created in the central catalog :



now that we have central catalog in place , we can run step 2 .

step 2 is where the consumer will request access to the data products in the catalog , he will do that by subscription model

asking for specific tables/databases and what type of premissions are required.

go to the mesh account , and in Cloud9 enviorment click on the left side on the **2\_consumer\_request\_access.py** file

is you try to run it , you will see it will ask for  - - database\_name , - -tables ,  - - request\_premissions

before entering the right parameters , we first set the enviorment variable same as we did in previous steps ,double clickiing on the 2\_consumer\_request\_access and then clicking on **ENV**

this is where we create an enviorment file which will point to our ceredntials file. set Name for **CredentialsFile** and Value to **/home/ec2-user/environment/Data-mesh-util/test/sample-test-creds.json**

now we can enter the relevant command , here is an example of how that should look like :

in this example we gave only select premisions to the consumer

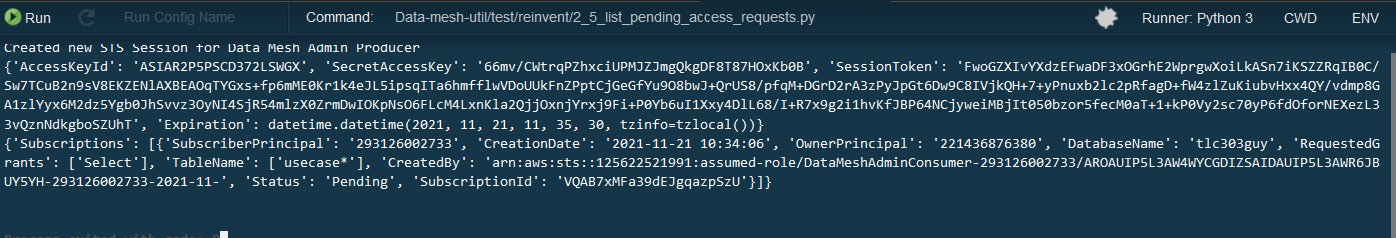
Data-mesh-util/test/reinvent/2\_consumer\_request\_access.py --database\_name tlc303guy --tables usecase\* —request\_permissions Select

the subscription will be stored in a dynamodb table.

you can now run  step 2\_5\_list\_pending\_access\_requests.py , before running it set up the **ENV** as you have done in previous steps.

this will help us see exactly what the pending requests are for the central account to aprove and provide premissions .

here is an example of the output of this step :



now , we are ready for step 3 , granting data product access.

in the cloud9 env , select and click **3\_grant\_data\_product\_access.py**

set the **ENV** file as done in previous steps.

for running the data prdouct access script you need to provide the following arguments are required: --subscription\_id, —grant\_permissions, —approval\_notes.

the subscription id can be retrieved from the output of step 2\_5\_list\_pending\_access\_requests , for grant premissions we will provide SELECT premissions .

please save the subscription id  , you will need to use it in step 4 as well .

here is an example of how the command should look like :

Data-mesh-util/test/reinvent/3\_grant\_data\_product\_access.py --subscription\_id FYbgaTuMXG63RnLsqaWv7m --grant\_permissions SELECT —approval\_notes approved

once completed , the mesh account provided access to the consumer for the data products.

next we will run the final step which is step number 4 . in step 4 the consumer will approve the resource sharing.

click on  left side  on **4\_finalize\_subscription\_and\_query.py**

set the **ENV** file as done in previous steps.

here is an example of how the command should look like :

Data-mesh-util/test/reinvent/4\_finalize\_subscription\_and\_query.py --subscription\_id YJapJ9GUcX5bmqT5fWnyC5

you can now log in to the consumer account and verify that the database and tables are seen in lakeformation.

you will see 2 databases , one is coming from the central mesh account , owned by the mesh account .

the other one is a resource link created and owned by the consumer account .  this allows the AWS data analytics and ML services to acess the data .

