

Start:

npm create-expo-app clonewhatsapp: create react folder

Dependencies:

1. npm install dayjs
2. npm install @react-navigation/native @react-navigation/native-stack
3. npm install @react-navigation/bottom-tabs
4. npm install -g @aws-amplify/cli

Some dependencies are in steps please follow that.

We have already created an whatsapp clone app, this document is for backend for the app
i.e. **AWS Amplify**

Some **AWS features we will be working with:**

Aws Amplify:

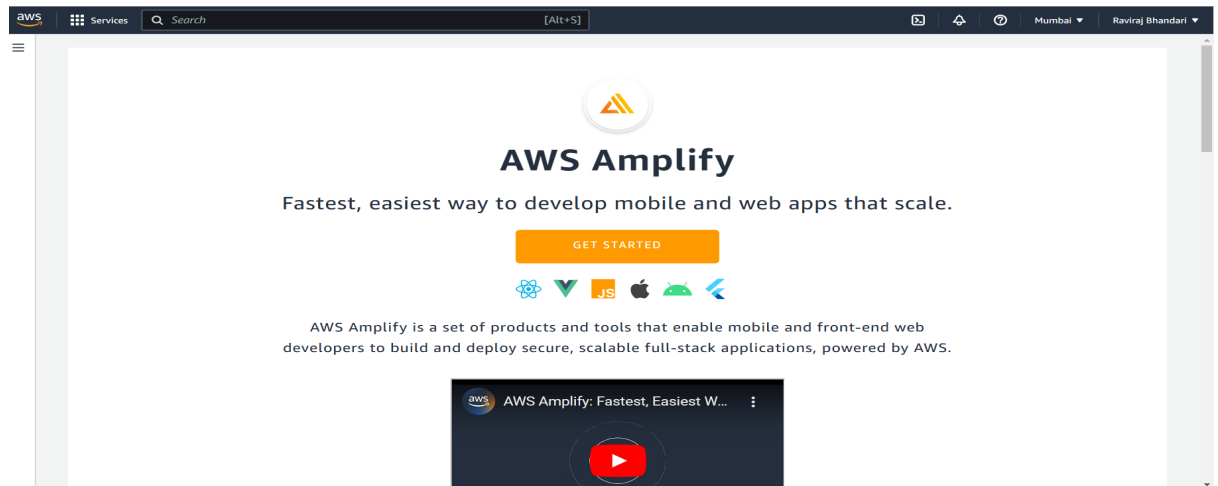
AWS Amplify is a set of products and tools that enable mobile and front-end web developers to build and deploy secure, scalable full-stack applications, powered by AWS.

Aws cognito**Aws appsync****Aws dynamodb****Aws graphql Api**

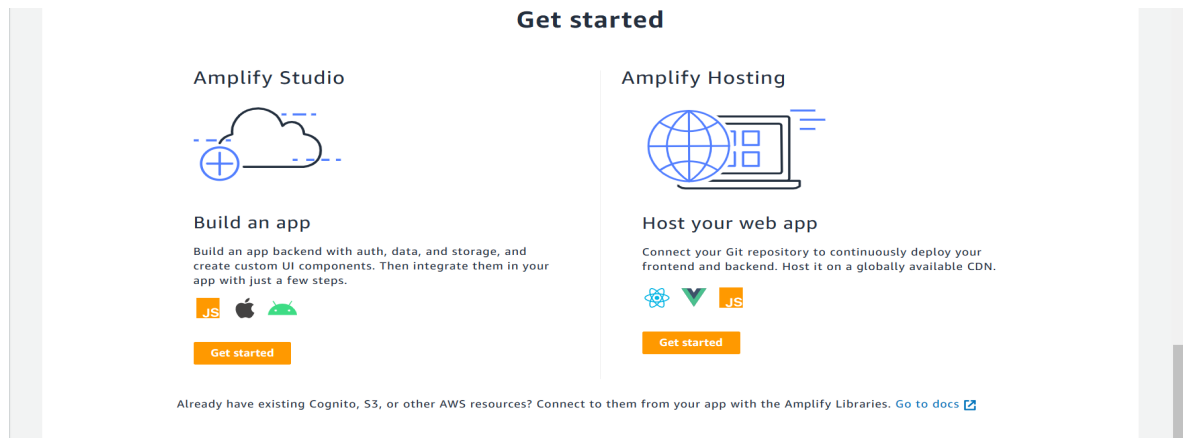
Setting Up Amplify:

Step1:

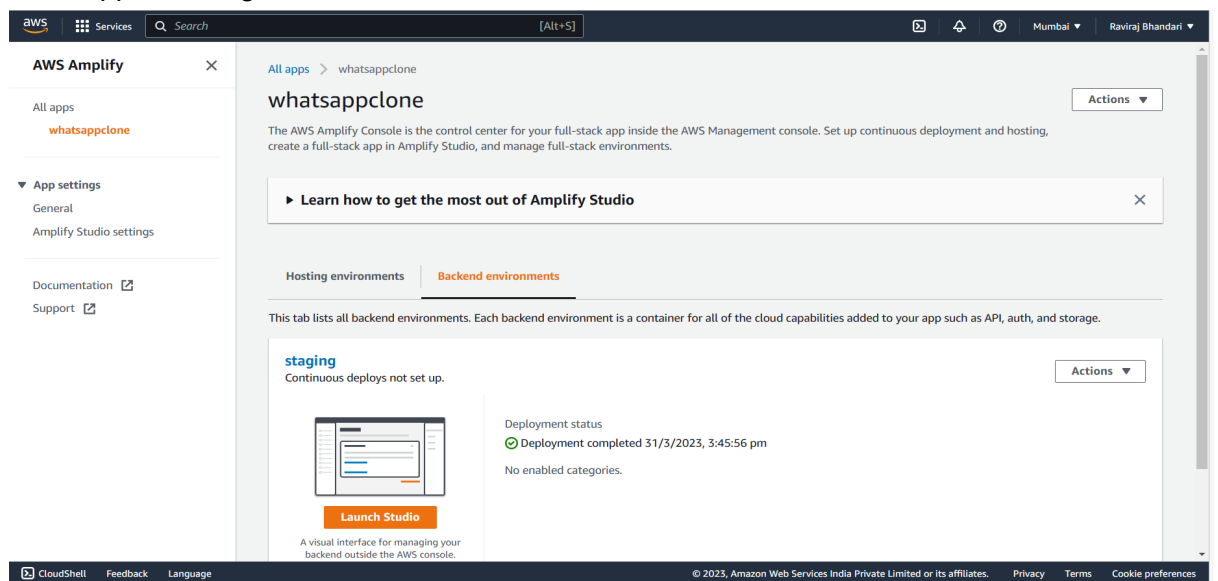
1. Login to AWS console
2. Search for **Amplify**



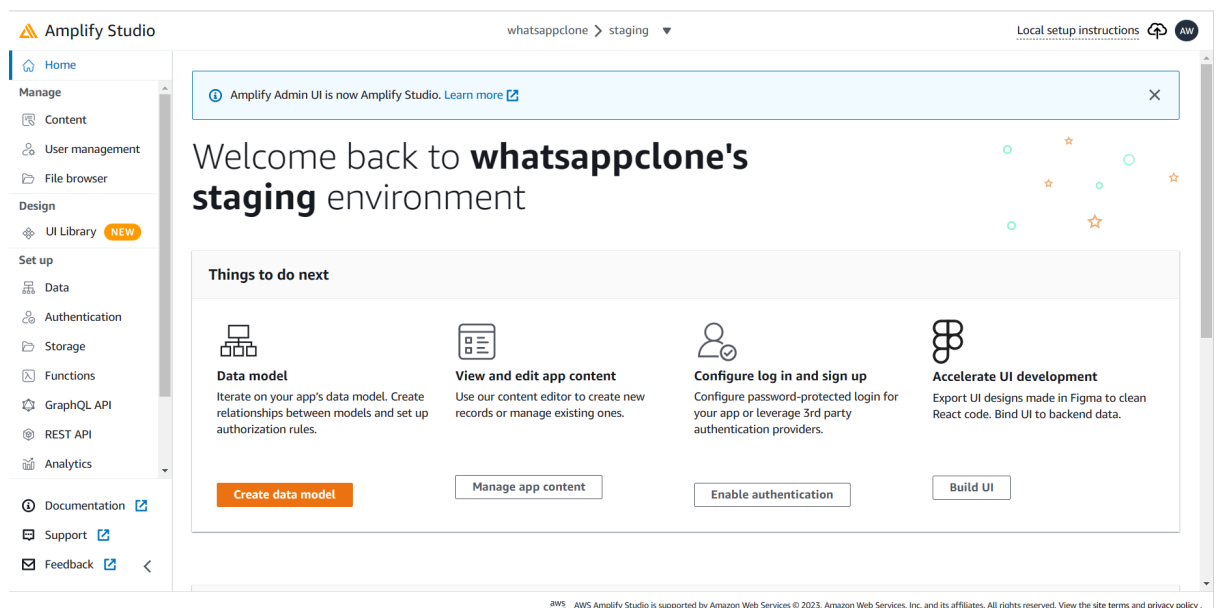
Scroll dow and select **build app**



3. Once app is build go to **launch studio**



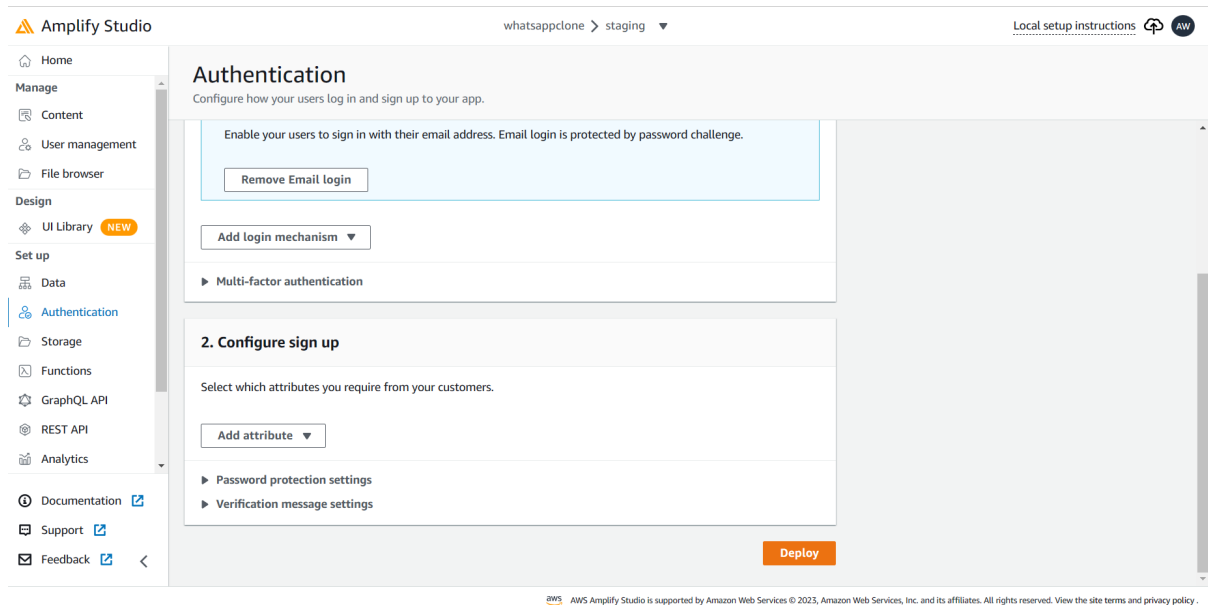
4. Here we can add **data** have **storage** and **authentication** options and also **users** and **content**



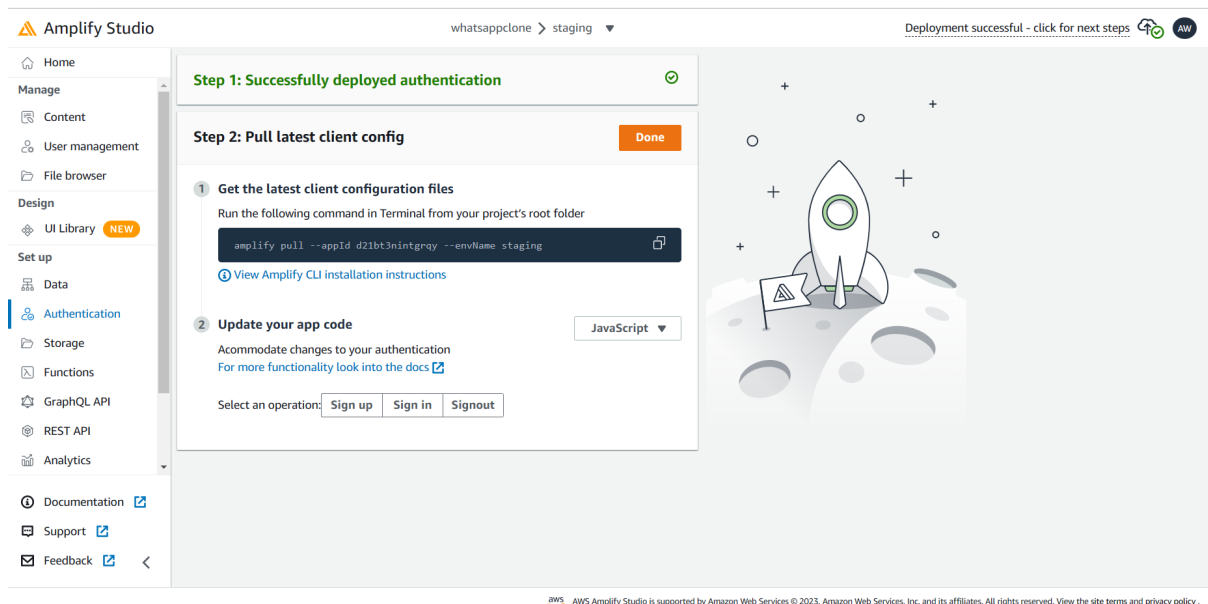
Step2: Working with the functions

1. Go to **set up**

- where we can **authenticate** the user by any means like google, amazon, facebook, gmail and phone number. *Here we use email*
- We can **add attributes** which will be required from your customers.
- Can also give **password** option for customers

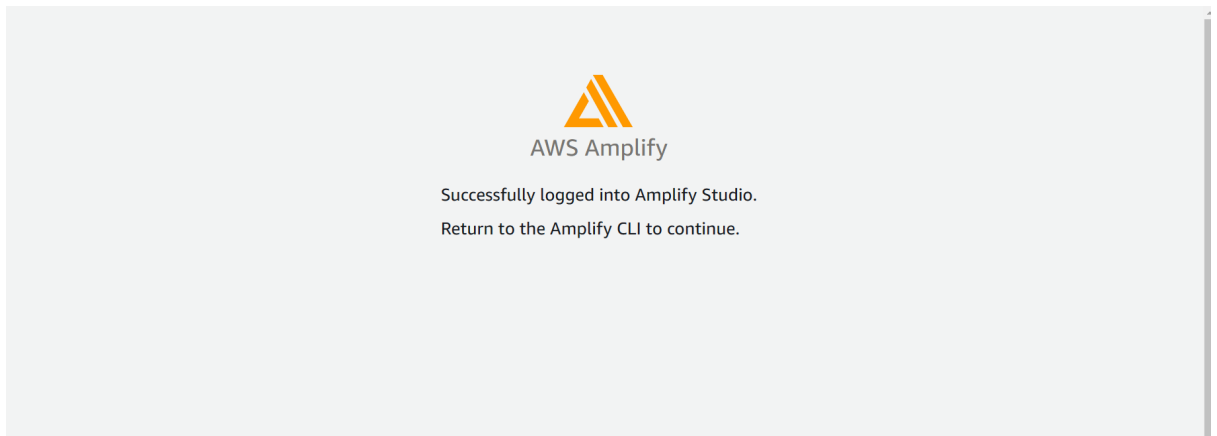


2. Once deployed we can pull our backend application into our react native project using credentials we received



Step3: connecting react with aws

- Go to vs and open terminal and copy the pull command and paste it in terminal,



2. After this in vs it will ask for some configuration
select : visual code, javascript, react-native, src, and default and yes

```
Successfully received Amplify Studio tokens.  
Amplify AppID found: d21bt3nintgrqy. Amplify App name is: whatsappclone  
Backend environment staging found in Amplify Console app: whatsappclone  
? Choose your default editor: Visual Studio Code  
? Choose the type of app that you're building (Use arrow keys)  
? Choose the type of app that you're building javascript  
Please tell us about your project  
? What javascript framework are you using react-native  
? Source Directory Path: src  
? Distribution Directory Path: /  
? Build Command: npm.cmd run-script build  
? Start Command: npm.cmd run-script start  
? Do you plan on modifying this backend? (Y/n) y
```

3. Now install some important dependencies by cmd:
npm install aws-amplify aws-amplify-react-native amazon-cognito-identity-js
@react-native-community/netinfo @react-native-async-storage/async-storage

Step4: Configure amplify into app

1. Add amplify into **App.js** which will help to **create login** and **signup** default page
2. Then also create an **Setting Screens** in **src** which will be for **signin**

Now we need our data to **amplify** so go to **aws amplify console**

Here we will be creating a page for users through which they can be the actual user of the app.

Now go to **Data** and create 3 models namely: **ChatRoom**, **Message**, **User** and we need to create a relationship between this models (refer images below)

Amplify Studio whatsappclone > staging Deployment

Home Manage Content User management File browser Design UI Library **NEW** Set up **Data** Authentication Storage Functions GraphQL API REST API Analytics Documentation Support Feedback

Data modeling

GraphQL API settings

+ Add model

ChatRoom

Field name	Type
id	ID!

+ Add a field

Relationship name	Related to	Cardinality
Messages	Message	1:n one ...
users	User	m:n man...
LastMessage	Message	1:1 one ...

+ Add a relationship SHIFT + ENTER

Message

Field name	Type
id	ID!
text	String!
chatroomID	Relationship Source
userID	Relationship Source

+ Add a field

+ Add a relationship

User

Field name	Type
------------	------

Amplify Studio whatsappclone > staging Deployment

Home Manage Content User management File browser Design UI Library **NEW** Set up **Data** Authentication Storage Functions GraphQL API REST API Analytics Documentation Support Feedback

Data modeling

GraphQL API settings

+ Add relationship SHIFT + ENTER

User

Field name	Type
id	ID!
name	String
image	String
status	String

+ Add a field

Relationship name	Related to	Cardinality
Messages	Message	1:n one ...
ChatRooms	ChatRoom	m:n man...

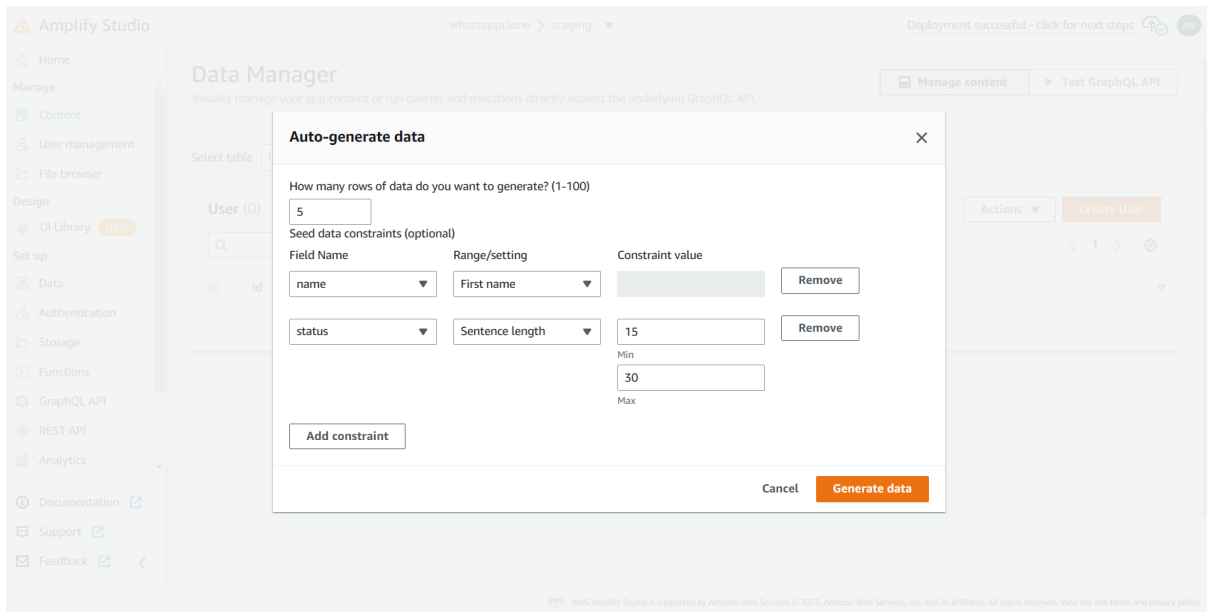
+ Add a relationship

Now **save and deploy**

Once done **copy the link and pull the data in the vs terminal**

Step5: Manage the content in Amplify

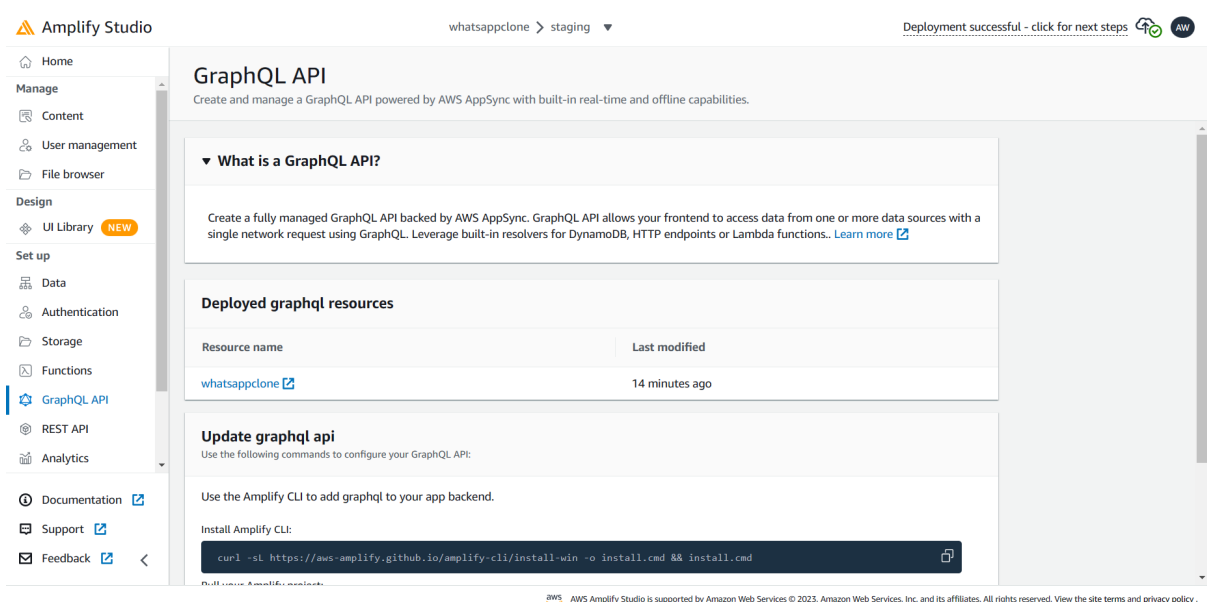
- Now select the user from drop down and select **User** and in **Action** generate some random users so that we can work with someone.



Now we will go for **GraphQL Api**

GraphQL Api:

Create a fully managed GraphQL API backed by AWS AppSync. GraphQL API **allows your frontend to access data from one or more data sources with a single network request using GraphQL**. Leverage built-in resolvers for DynamoDB, HTTP endpoints or Lambda functions.



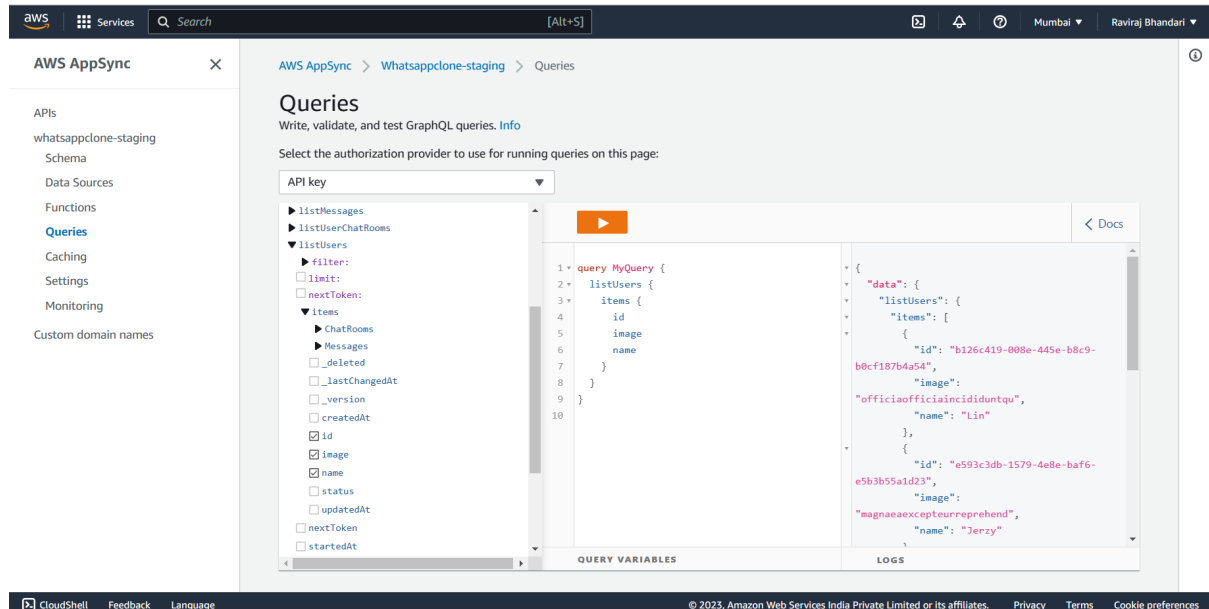
Now as from above image go to resource name and **click on app** This will now relocate us to the **AppSync**

AppSync:

AppSync allows developers to build GraphQL APIs without much of the usual work; it **handles the parsing and resolution of requests as well as connecting to other AWS**

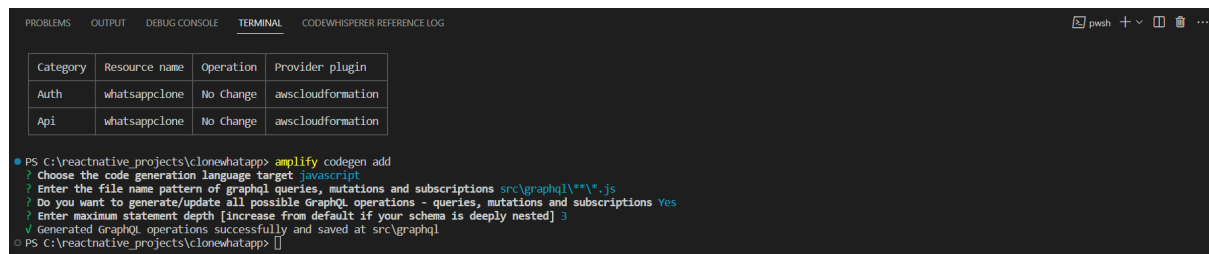
services like AWS Lambda, NoSQL and SQL data stores, and HTTP APIs to **gather backend data for the API**.

Now we can see some features we can edit and work with like schema and queries, see image below:



Here let's go to query and **add id, image, name**, now let's run this query in our VS application

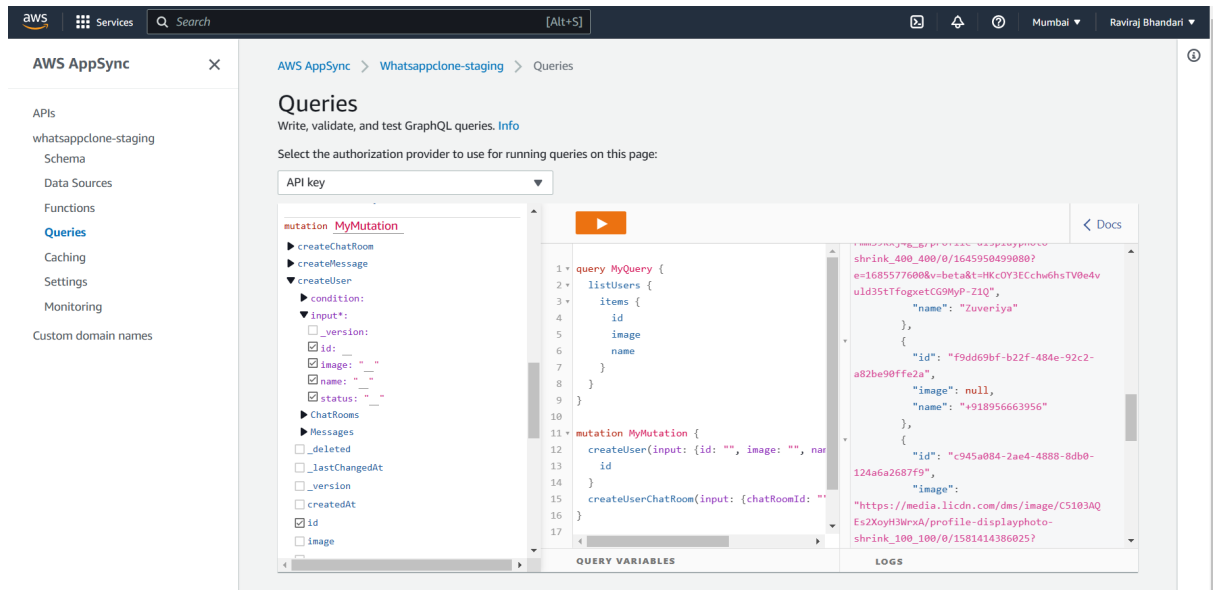
Now in terminal type: **amplify codegen add** and follow the steps in image below:



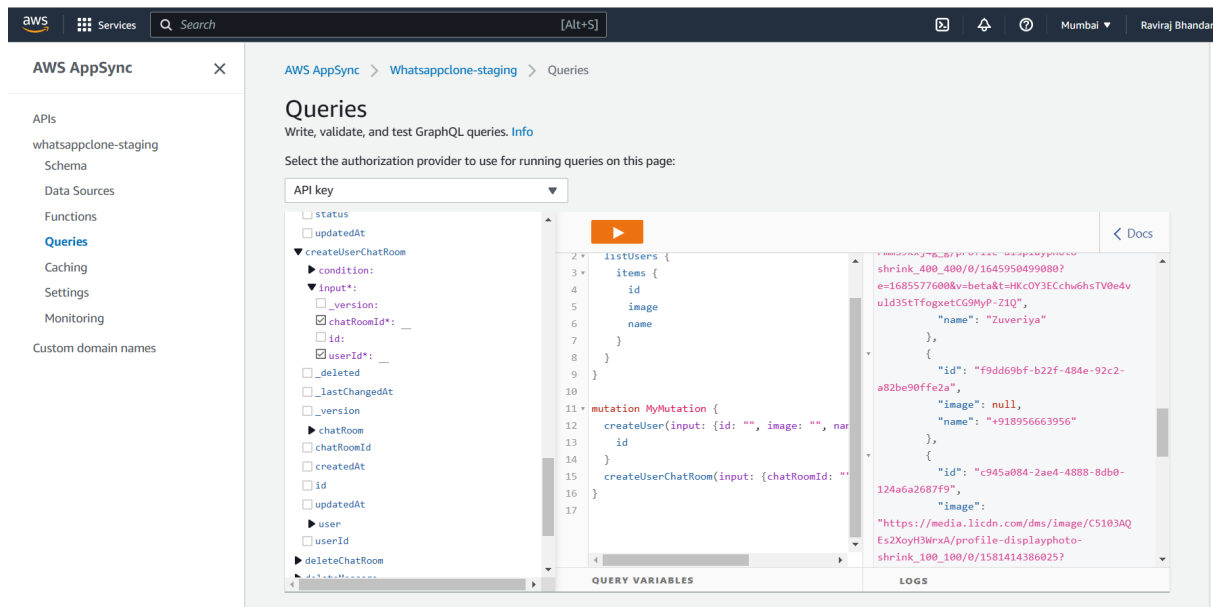
Now we need to auth the user with the database

So we will connect aws with **app.js** (refer code on github): **link**:

Go to **queries** and then their scroll down **mutation** and refer the following image for changes



Now go to query for the same and edit according to the image:



AWS AppSync

APIs

whatsappclone-staging

Schema

Data Sources

Functions

Queries

Caching

Settings

Monitoring

Custom domain names

AWS AppSync > Whatsappclone-staging > Queries

Queries

Write, validate, and test GraphQL queries. Info

Select the authorization provider to use for running queries on this page:

API key

Query

query MyQuery {
 getChatRoom
 getMessage
 getUser {
 id: f9dd69bf-b22f-484
 ChatRooms {
 filter:
 limit:
 nextToken:
 sortDirection:
 items {
 _deleted
 _lastChangedAt
 _version
 chatRoom
 chatRoomId
 createdAt
 id
 }
 }
 }
}

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

query MyQuery {
 listUsers {
 items {
 id
 image
 name
 }
 }
 getUser(id: "f9dd69bf-b22f-484e-92c2-a82l") {
 id
 name
 ChatRooms {
 items {
 id
 }
 }
 }
}

zimage :
"https://media.lidn.com/dms/image/D4D03AQ
Gf83T786A3iw/profile-displayphoto-
shrink_400_400/0/1672066281429?
e=1685577600&v=beta&t=Km4rffjny9PFk28aniTG
_zlm4qni4LlwFH3nV6mAUE",
"name": "Raviraj"
}
},
"getUser": {
 "id": "f9dd69bf-b22f-484e-92c2-
a82be90ffe2a",
 "name": "+918956663956"
}
}
}

QUERY VARIABLES

LOGS

AWS AppSync

APIs

whatsappclone-staging

Schema

Data Sources

Functions

Queries

Caching

Settings

Monitoring

Custom domain names

AWS AppSync > Whatsappclone-staging > Queries

Queries

Write, validate, and test GraphQL queries. Info

Select the authorization provider to use for running queries on this page:

API key

Messages {
 _deleted
 _lastChangedAt
 _version
 chatRoomLastMessageId
 createdAt
 id
 updatedAt
 users {
 filter:
 limit:
 nextToken:
 sortDirection:
 items {
 nextToken
 startedAt
 }
 }
 chatRoomId
 createdAt
 id
 updatedAt
}

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

name
}
}
getUser(id: "f9dd69bf-b22f-484e-92c2-a82l") {
 id
 ChatRooms {
 items {
 chatRoom {
 id
 users
 }
 }
 }
}
}
}
mutation MyMutation {
}

SHRINK_400_400/0/10720000001429?
e=1685577600&v=beta&t=Km4rffjny9PFk28aniTG
_zlm4qni4LlwFH3nV6mAUE",
"name": "Raviraj"
}
},
"getUser": {
 "id": "f9dd69bf-b22f-484e-92c2-
a82be90ffe2a",
 "name": "+918956663956",
 "ChatRooms": {
 "items": []
 }
}
}

QUERY VARIABLES

LOGS

AWS AppSync

APIs

whatsappclone-staging

Schema

Data Sources

Functions

Queries

Caching

Settings

Monitoring

Custom domain names

AWS AppSync > Whatsappclone-staging > Queries

Queries

Write, validate, and test GraphQL queries. Info

Select the authorization provider to use for running queries on this page:

API key

items {
 _deleted
 _lastChangedAt
 _version
 chatRoom {
 lastMessage {
 _deleted
 _lastChangedAt
 _version
 chatRoomId
 createdAt
 id
 text
 updatedAt
 userID
 }
 Messages {
 _deleted
 _lastChangedAt
 _version
 chatRoomLastMessageId
 }
 }
}

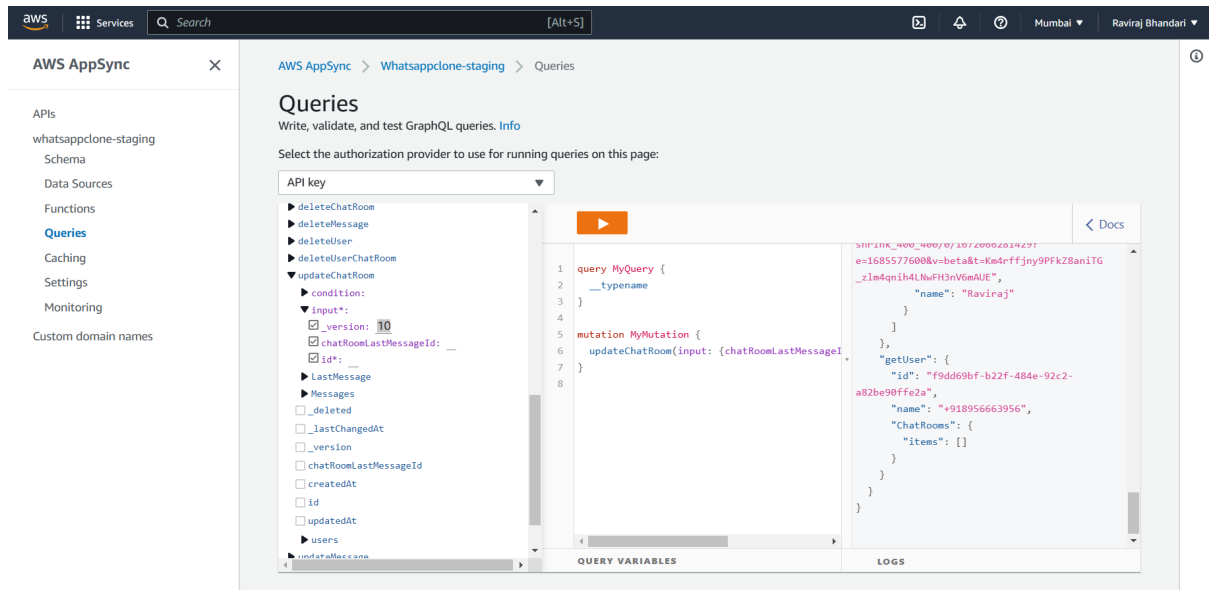
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

query MyQuery {
 listUsers {
 items {
 id
 image
 name
 }
 }
 getUser(id: "f9dd69bf-b22f-484e-92c2-a82l") {
 id
 ChatRooms {
 items {
 chatRoom {
 id
 users
 lastMessage {
 id
 }
 }
 }
 }
 }
}

SHRINK_400_400/0/10720000001429?
e=1685577600&v=beta&t=Km4rffjny9PFk28aniTG
_zlm4qni4LlwFH3nV6mAUE",
"name": "Raviraj"
}
},
"getUser": {
 "id": "f9dd69bf-b22f-484e-92c2-
a82be90ffe2a",
 "name": "+918956663956",
 "ChatRooms": {
 "items": []
 }
}
}

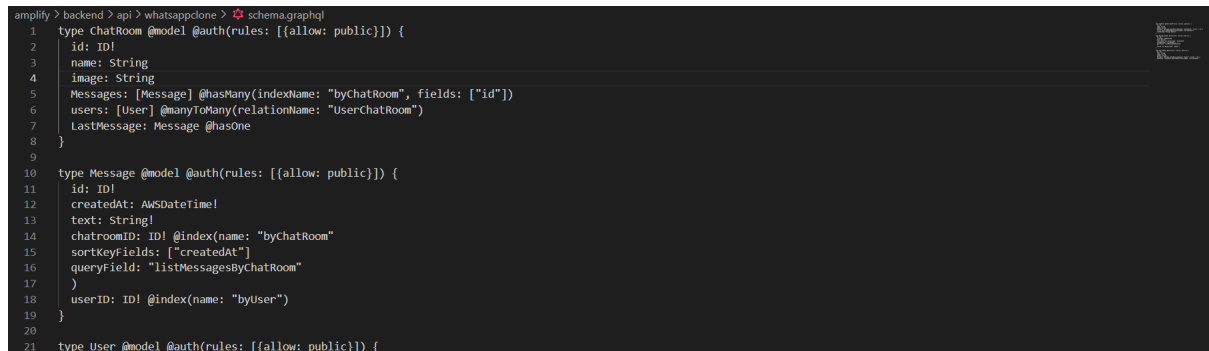
QUERY VARIABLES

LOGS



Now Will look for sorting the messages, here **graphQL** will hep to create an abstract layer around the sorting this is called index.

Now in vs code in Amplify go to api then schema.graphql and edit the following (refer img)



Then in terminal **amplify push** this will update the amplify in aws console
It will then ask some question put **yes** to all

Queries: are to get data

Mutation: update data on server

Subscription: to receive updates from server