* **Setting up DocumentDB**:
  + Navigate to the Amazon DocumentDB console.
  + Create a new cluster with the necessary configurations.
  + Note down the connection string for future use.
* **Lambda & API Gateway**:
  + Navigate to the AWS Lambda console and create a new function.
  + Use the provided Python code for the function.
  + In API Gateway, create a new API and add a POST endpoint for taxi location updates.
  + Link the API endpoint to the Lambda function.
* **Taxi Simulator**:
  + Run the provided Python script on your local machine or an EC2 instance.
  + Ensure the API\_URL variable in the script points to the API Gateway endpoint you created.
* **Testing**:
  + After starting the taxi simulator, you should see location updates being sent to the API.
  + Check the DocumentDB collection to ensure that taxi locations are being updated correctly.
* **User and Taxi Registration**:
  + Navigate to the AWS Lambda console and create new functions for user and taxi registrations.
  + Use the provided Python code for the functions.
  + In API Gateway, create new API endpoints for user and taxi registrations.
  + Link the API endpoints to the respective Lambda functions.
* **User Taxi Request**:
  + Navigate to the AWS Lambda console and create a new function for user taxi requests.
  + Use the provided Python code for the function.
  + In API Gateway, create a new API endpoint for user taxi requests.
  + Link the API endpoint to the Lambda function.
* **Testing**:
  + Use tools like Postman to test the user and taxi registration API endpoints.
  + Test the user taxi request endpoint by sending a request with user details and location. Check the response to see the nearest available taxis.
* **Trip Fulfillment**:
  + Navigate to the AWS Lambda console and create a new function for trip fulfillment.
  + Use the provided Python code for the function.
  + In API Gateway, create a new API endpoint for trip fulfillment.
  + Link the API endpoint to the Lambda function.
* **Real-time Visualization**:
  + Set up a Kinesis stream in the AWS Kinesis console.
  + Create a Lambda function to process the Kinesis stream.
  + In QuickSight, create a new analysis and add a map visualization.
* **Analytics**:
  + Set up Athena in the AWS Athena console.
  + In QuickSight, create a new analysis and add visualizations based on Athena queries.
* **Taxi Distribution Hints**:
  + Navigate to the AWS Lambda console and create a new function for sending taxi distribution hints.
  + Use the provided Python code for the function.
  + Schedule the function to run at regular intervals using CloudWatch Events.