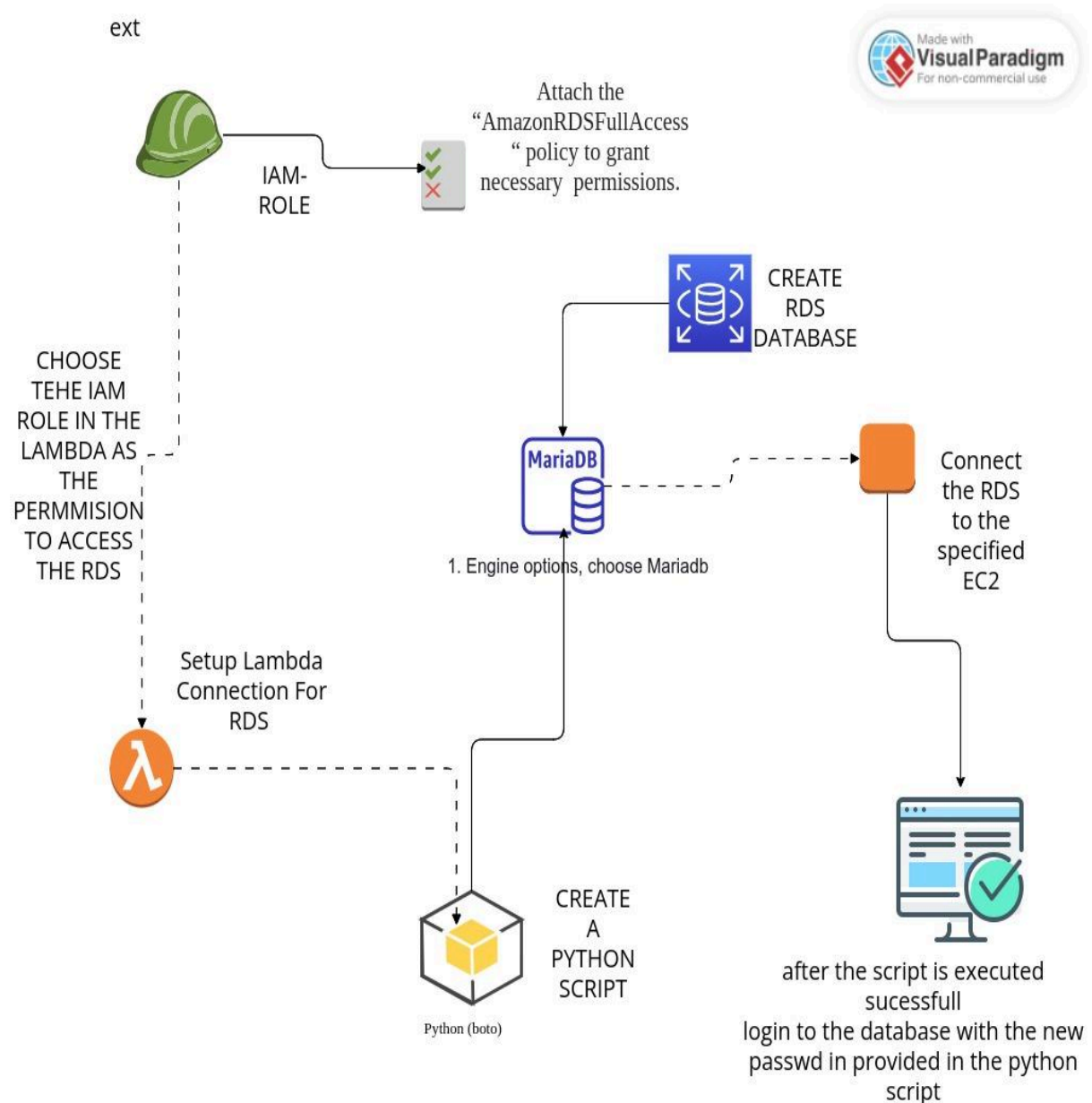


Using a Lambda function to change an Amazon RDS passwd



1] CREATE IAM ROLE

Create an IAM role Attach the "AmazonRDSFullAccess" policy to grant necessary permissions.

2]CREATE RDS DATABASE

To create a database instance

1. Open the Amazon RDS console and choose Create database.


Choose a database creation method [Info](#)


☐ **Standard create**
You set all of the configuration options, including ones for availability, security, backups, and maintenance.


☒ **Easy create**
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.


Configuration


Engine type [Info](#)


☐ **Aurora (MySQL Compatible)**


☐ **Aurora (PostgreSQL Compatible)**


☒ **MySQL**


☐ **MariaDB**


☐ **PostgreSQL**


☐ **Oracle**


2. Leave the Standard create option selected, then in Engine options, choose Mariadb
3. In Templates, choose Free tier.
4. In Settings, for DB instance identifier, enter “your identifier name”.
5. Set your username and password by doing the following:

Master password [Info](#)

.....

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password [Info](#)


.....

► **Set up EC2 connection - *optional***

You can also set up a connection to an EC2 instance after creating the database. Go to the database list page or the database details page, choose **Actions**, and then choose **Set up to EC2 connection**.

► **View default settings for Easy create**

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use [Standard create](#).

 You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel
Create database

- In Credentials settings, leave Master username set to `admin`.
- For Master password, enter and confirm a password to access your database.

6. Connect the RDS to the specified EC2

7. Leave all the remaining default options selected and choose Create

Setup Lambda Connection For RDS

Create the function

1] On this page select the execution environment as **Python**

- Install **boto3** and **required** package for the code to accessing DB.
- Make it as a zip file and upload the zip to the lambda function

Upload a .zip file

When you upload a new .zip file package, it overwrites the existing code.

Upload

RDSConnection.zip
331.97 KB

For files larger than 10 MB, consider uploading using Amazon S3.

Cancel

Save

- **Test The Function For Access**
- Click on Test from code editor of lambda function. It will open event creation page.

Give any name to the event and leave everything as default and proceed.

Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event

☐ Edit saved event

Event name

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

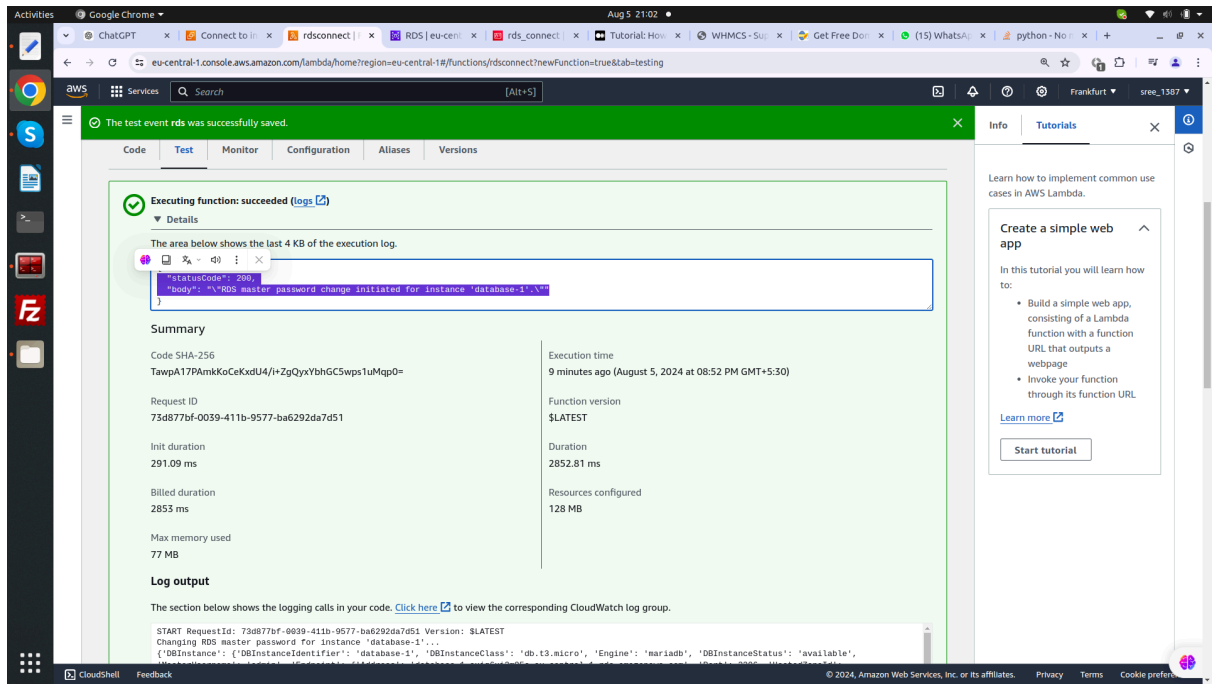
Template - optional

Cancel

Invoke

Save

- Once the event is created click on Test.



Python script

```
import boto3
import json
```

```
def change_rds_password(db_instance_identifier, new_password):
    # Create an RDS client using the IAM role credentials
    rds_client = boto3.client('rds')

    try:
        # Modify the RDS instance to update the master user password
        response = rds_client.modify_db_instance(
            DBInstanceIdentifier=db_instance_identifier,
            MasterUserPassword=new_password,
            ApplyImmediately=True # Apply the changes immediately
```

```

    )

    print(f"Changing RDS master password for instance
    '{db_instance_identifier}'...")
    print(response)
    return {
        'statusCode': 200,
        'body': json.dumps(f"RDS master password change initiated for
    instance '{db_instance_identifier}'.")
    }
except Exception as e:
    print(f"Error changing RDS master password: {e}")
    return {
        'statusCode': 500,
        'body': json.dumps(f"Error changing RDS master password:
    {str(e)}")
    }

def lambda_handler(event, context):
    # Define your RDS instance details
    db_instance_identifier = 'database-1'
    new_password = 'admin1234'

    return change_rds_password(db_instance_identifier, new_password)

if __name__ == "__main__":
    # For local testing, call the lambda_handler function with dummy
    event and context
    lambda_handler({}, {})

```

