Prerequisites:

- Install and set up AWS CLI on your local machine.
- Install Docker Desktop.
- Install SAM CLI (Serverless Application Model Command Line Interface).
- Have Python and pip installed on your local machine.
- Have PostgreSQL installed and configured locally or have access to a remote PostgreSQL database.

Setting up the Environment:

- Create a new directory for your project and navigate to it using the terminal.

Initialize a new SAM project:

- Run the following command to create a new SAM project:
  ```
  sam init
  ```
- Choose the "AWS Quick Start Templates" - option.
- Select the "1 - Hello World Example" template (for simplicity in this example).

Develop Lambda Function and Test Locally:

- Navigate to the "hello_world" directory inside your project directory.
- Modify the app.py file to write your Lambda function in Python.

After modifying app.py - Build Sam:

- After performing modification/Changes in app.py – make sure you saved (Ctrl + S)
- Run following command to build sam file:
  ```
  sam build
  ```

Testing the Lambda Function Locally:

- Run the following command to test the function locally:
  ```
  sam local invoke HelloWorldFunction –events/event.json
  ```
- Ensure your function works as expected.

Configuring Docker for Local Testing:

- SAM CLI can leverage Docker for a Lambda-like execution environment. This helps you simulate AWS Lambda locally.
- Make sure Docker Desktop is running on your machine.

Externalizing API Endpoints and Credentials:

- Create a configuration file (e.g., config.json) to store all your API endpoints and credentials.
- Load the configurations in your Python code using configparser or other appropriate libraries.
- Never hard-code sensitive information in your code.

Create AWS CloudFormation Scripts:

- Create a CloudFormation template (template.yml) to define your serverless application's infrastructure.
- Define AWS resources like Lambda function, API Gateway, IAM roles, etc., in the CloudFormation template.
- Use the AWS::Serverless::Function resource type to define your Lambda function.
- You can reference environment variables, API endpoints, and other configurations from the configuration file in your CloudFormation template.

Deploy the Application:

- Use the following command to deploy your SAM application to AWS:
  ```
  sam deploy --guided
  ```
- The --guided flag will prompt you for configurations like stack name, AWS Region, etc.

Testing the Deployed Application:

- After successful deployment, test your Lambda function and the API endpoints using the AWS Management Console.