👚Serverless E-Commerce Backend

A real-world, scalable **e-commerce backend** built using **AWS Serverless Architecture** and **Python**. This project showcases how modern cloud-native applications can be developed without managing servers, while maintaining high availability and cost-efficiency.

🗜Features

- 🔐**User Authentication** (JWT/Cognito-based)
- 📦**Product Catalog Management**
- 🛍️**Order Placement & Tracking**
- **Inventory Management**
- 📊**Logging & Monitoring with CloudWatch**
- 🩲**RESTful API Gateway Integration**

🧱Architecture

- **API Gateway**: Handles incoming HTTP requests
- **AWS Lambda**: Backend logic for products, orders, and users
- **Amazon DynamoDB**: NoSQL database for fast and flexible storage
- **Amazon S3**: Stores product images or static files
- **AWS Cognito (Optional)**: Secure user sign-up, sign-in, and access control
- **IAM Roles**: Fine-grained permissions for services
- **CloudWatch**: Real-time logs and monitoring

🛠️Tech Stack

- **Language**: Python 3.x
- **Cloud Provider**: AWS
- **Database**: Amazon DynamoDB
- **API Management**: Amazon API Gateway
- **Function Execution**: AWS Lambda
- **Authentication**: AWS Cognito or JWT
- **Monitoring**: AWS CloudWatch

📂Project Structure

```
serverless-ecommerce-backend/
├── products/
```

```
│   └── handler.py
├── orders/
│   └── handler.py
├── users/
│   └── auth.py
├── utils/
│   └── db_helper.py
├── template.yaml         # AWS SAM or Serverless Framework template
├── requirements.txt
└── README.md
```

🧸 Testing

- Unit tests for each Lambda function
- API testing via **Postman** or **cURL**
- Logging enabled for all endpoints using **CloudWatch Logs**

💡 Learning Outcomes

- Real-world experience in **Serverless Architecture**
- Built scalable microservices without managing infrastructure
- Learned practical use of **AWS IAM, Lambda, DynamoDB, S3**, and **API Gateway**
- Gained exposure to **Python-based cloud functions**

🔗 Future Enhancements

- Integrate **payment gateway** (Razorpay/Stripe)
- Admin dashboard for inventory updates
- Email/SMS notifications using **SNS**
- CI/CD pipeline with **AWS CodePipeline**

📚 Resources Used

- AWS Documentation
- AWS Cloud Practitioner Learning Path
- Serverless Stack (SST) Guides
- Medium articles and GitHub templates

💻Author

Omkar\ Feel free to connect on LinkedIn | GitHub | Portfolio

📜License

This project is licensed under the MIT License.