Instructions:
1. You should use infrastructure as code to deploy all Lambda functions and needed infrastructure. You may use AWS CDK, Cloudformation, AWS SAM or Terraform.
2. You should use SQS or EventBridge to queue payment events.
3. You should use DynamoDB to store order and payment data, one table per service, and you must follow a single table design for both microservices.
4. You must use two lambdas for each service, one for handling events and one for handling requests from API Gateway.
5. You should use the AWS SDK for Golang to interact with DynamoDB and SQS.
6. You should use API Gateway to trigger the functions to create orders and receive payments.
7. When an order is created, the Orders service should send an orderCreatedEvent event to the Payments service via SQS or EventBridge with an incomplete status.
8. When a payment is received, the Payments service should update the payment event status to complete in DynamoDB and send an order completed event to the Orders service via SQS or EventBridge.
9. When an order is completed, the Orders service should mark the order as ready for shipping in DynamoDB.
10. You should write unit tests for your functions.
11. You should handle errors gracefully.