1. Introduction
   1. Let's Say You Have an E-Commerce Company
   2. You created 3 Lambda functions for different tasks
      1. Lambda 1 — Checks if the payment is successful
      2. Lambda 2 — Updates the order and prepares it for shipping
      3. Lambda 3 — Sends a confirmation email to the customer
   3. Now the question is
      1. How do you connect these 3 Lambda functions in a proper flow?
      2. How to make sure Lambda 2 runs only after Lambda 1, and so on?
      3. What if payment fails? You want to stop the flow or handle it differently, right?
   4. The Answer is AWS Step Functions
      1. AWS Step Functions is a serverless service that helps you connect and manage multiple tasks in a proper flow
      2. It lets you build a state machine – a step-by-step diagram – where each step performs a specific action like checking payment, updating orders, or sending emails
      3. You can control the order of steps, make decisions based on results, and handle failures or retries automatically — all without writing complex code
      4. Even if something fails, it can take an alternate route or retry as per your logic, based on how you define the state machine
2. What is a State Machine?
   1. A state machine is a type of workflow where you define steps (called states) and how they should run one after another
   2. It controls the flow of logic — like what should happen first, what happens next, and what to do if something fails
   3. You can create a state machine in two ways
      1. Using JSON or YAML (manual way)
      2. Using AWS Console's visual workflow designer (GUI) – which is drag-and-drop and beginner-friendly
   4. Example (Simple Flow) That complete flow = one state machine.
      1. Start
      2. Call Lambda to check payment
      3. If success → prepare order
      4. Then → send email
      5. End
2. Why Use Step Functions?
   1. Helps connect multiple services like Lambda, ECS, SQS
   2. Reduces complex code — use flow instead of nested if-else
   3. Visual workflow – easy to monitor and debug
   4. Error handling and retry support built-in
   5. Good for automation and microservice communication