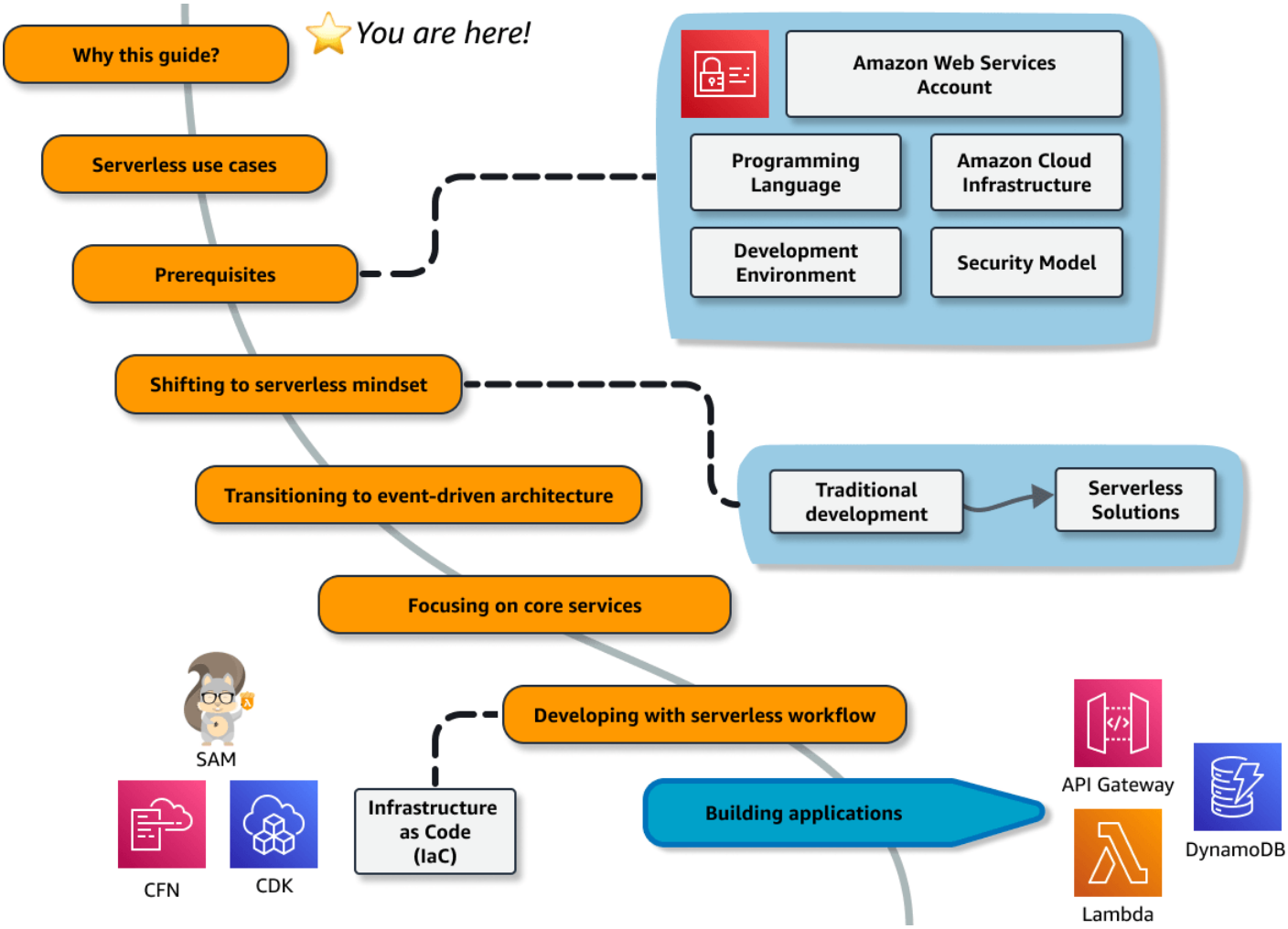
# What is serverless development?

The following topics will guide you through developing a better conceptual understanding of serverless application development, and how various AWS services fit into together to create *application patterns* that form the core of your cloud applications. These applications can range from microservices that handle discreet business logic as a part of your application back-end, to event-driven workflows that perform data transformations or processing.

Understanding serverless development will you help you make critical decisions about which AWS services are best suited for your business need. For example, choosing between Amazon DocumentDB, DynamoDB, and Aurora PostgreSQL for a database depends on various factors, such as what type of data-structure you want to use, or how many concurrent database connections you anticipate as your applications scale.

The goal of this serverless developer guide is to give you directed **learning paths** for the core services you need to implement serverless solutions.

Serverless development lets you build applications without managing long-running servers, such as a provisioned Amazon EC2 instance. AWS serverless technologies are pay-as-you-go, can scale up and down as your application needs change, and are built to expand across AWS Regions to ensure resiliency.




This guide will highlight what you need to know right away and link to service documentation for more service-specific details.

For example, you will learn that the Lambda service creates an *execution environment* to run compute functions. For more information on how Lambda manages function scaling or reduces start-up time, we will link you to relevant sections of the Lambda developer guide.

The topics in this guide will cover the prerequisites for understanding serverless development on AWS, such as account creation and an overview of AWS cloud infrastructure. Then, you will learn how to shift from a traditional development model to a serverless, event-driven architecture with which to develop applications on the cloud.

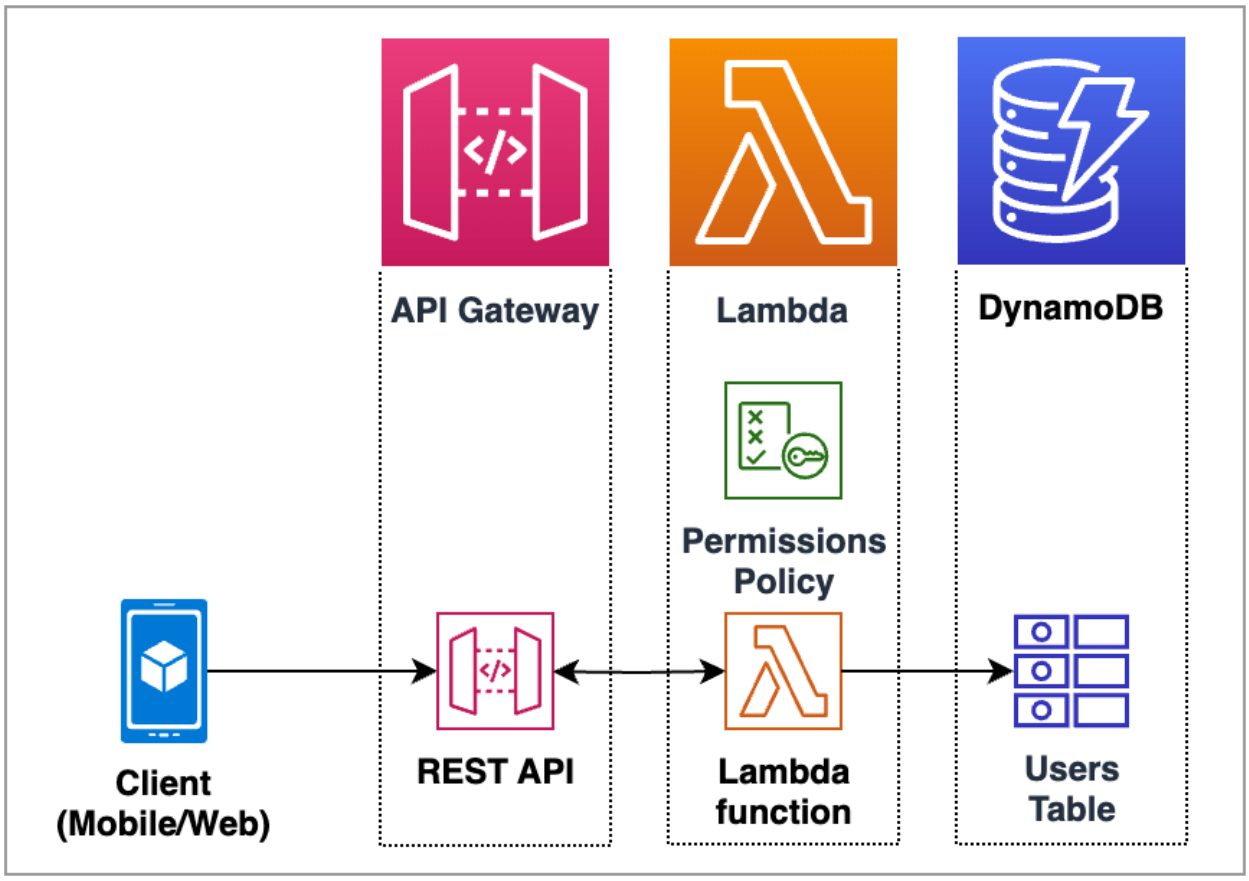
Along the way, this guide will introduce core services, workshops, and tutorials, you can choose to reinforce your learning with hands-on activities.

- AWS Identity and Access Management — for securely accessing resources on AWS.

- AWS Lambda — for serverless compute functionality.
- Amazon API Gateway for integrating HTTP and HTTPS requests with services to handle the requests.
- Amazon DynamoDB for data storage and retrieval

**Learn serverless techniques in an online workshop**

Learn by doing in the Serverless Patterns Workshop. The first module introduces a serverless microservice to retrieve data from DynamoDB with Lambda and API Gateway. Additional modules provide practical examples of unit and integration testing, using infrastructure as code to deploy resources, and how to build common architectural patterns used in serverless solutions.



# Understanding serverless data processing

Processing data in serverless applications largely falls within the following three patterns:

- **Asynchronous processing**– big data processing, image/video manipulation, web hooks
- **Synchronous processing** – web apps, web services, microservices, web hooks
- **Streaming** – processing inbound data streams, from apps, IoT devices

The following topics provide a broad overview of each serverless processing pattern and explain the most common services you can use for each type. Use these topics to gain a conceptual understanding of serverless data processing on AWS.

**Topics**



# Asynchronous processing

Serverless development allows your applications to ingest, process and analyze high volumes of data quickly and efficiently.

As the volume of data coming from increasingly diverse sources grows, you might find you need to move quickly to process this data to ensure that your application's business logic can meet your needs. To process data at scale, organizations need to elastically provision resources to manage the information they receive from various microservices, mobile devices, operational data stores, and other sources.

Learn how to build a scalable serverless data processing solution. Use Amazon Simple Storage Service to trigger data processing or load machine learning (ML) models so that Lambda can perform ML inference in real time.

- **File processing** – Suppose you have a photo sharing application. People use your application to upload photos, and the application stores these user photos in an Amazon S3 bucket. Then, your application creates a thumbnail version of each user's photos and displays them on the