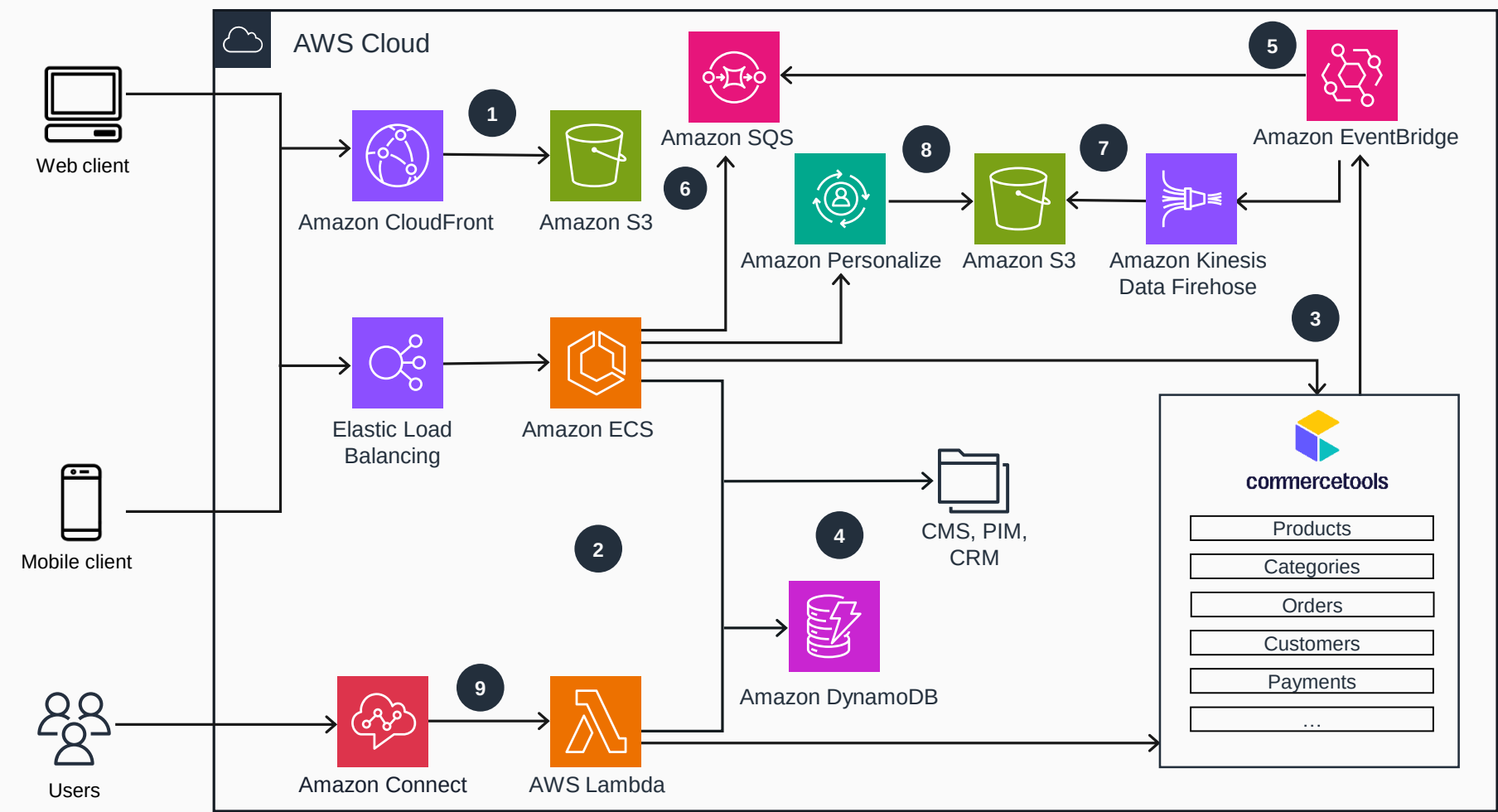


Guidance for Building an Ecommerce Experience with Commercetools on AWS

This architecture diagram shows how to integrate AWS services with commercetools so you can build an ecommerce experience using the MACH approach.



- 1 Host the static parts of your brand-specific frontend (such as web pages and images) on **Amazon Simple Storage Service (Amazon S3)**, delivered through **Amazon CloudFront** with low latency.
- 2 Your business logic runs in containers using **Amazon Elastic Container Service (Amazon ECS)** or **AWS Lambda** to run your code in serverless functions.
- 3 Integrate commercetools APIs with your business logic to gain flexibility when composing your shopping experience.
- 4 Store additional business data in databases (SQL or NoSQL) or integrate other systems in your IT landscape, such as a content management system (CMS), product information management (PIM) system, or customer relationship management (CRM) system.
- 5 Integrate asynchronously with events from commercetools (such as new orders) and use **Amazon EventBridge** to distribute those events to your applications.
- 6 **Amazon Simple Queue Service (Amazon SQS)** receives events from **EventBridge** and queues them for processing by downstream services.
- 7 By storing commercetools events in an **Amazon S3**-powered data lake ingested through **Amazon Kinesis Data Firehose**, you can make events accessible for visualizations, forecasting, and analyzing usage patterns.
- 8 Use data such as previous buying behavior to tailor customers' shopping experience with **Amazon Personalize**.
- 9 Use **Amazon Connect** as your contact center to allow your customers to connect with you through phone or chat. By accessing the same business logic and data in commercetools, you can provide a consistent and unified experience for customers.