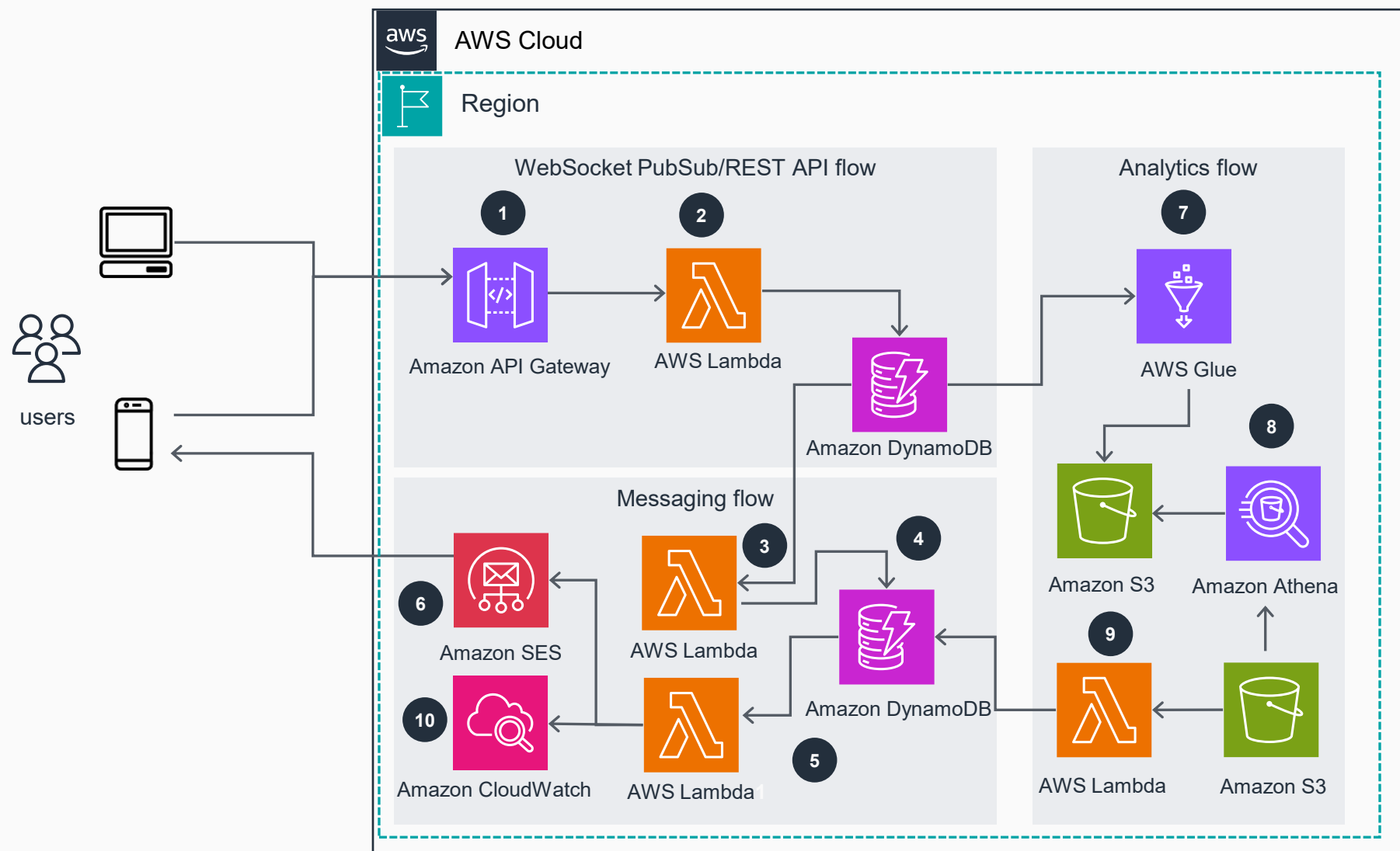


Guidance for Games Messaging System on AWS

This architecture shows a flexible gaming messaging system that delivers real-time notifications for friend activities and targeted marketing campaigns. This serverless architecture reduces costs and scales automatically.

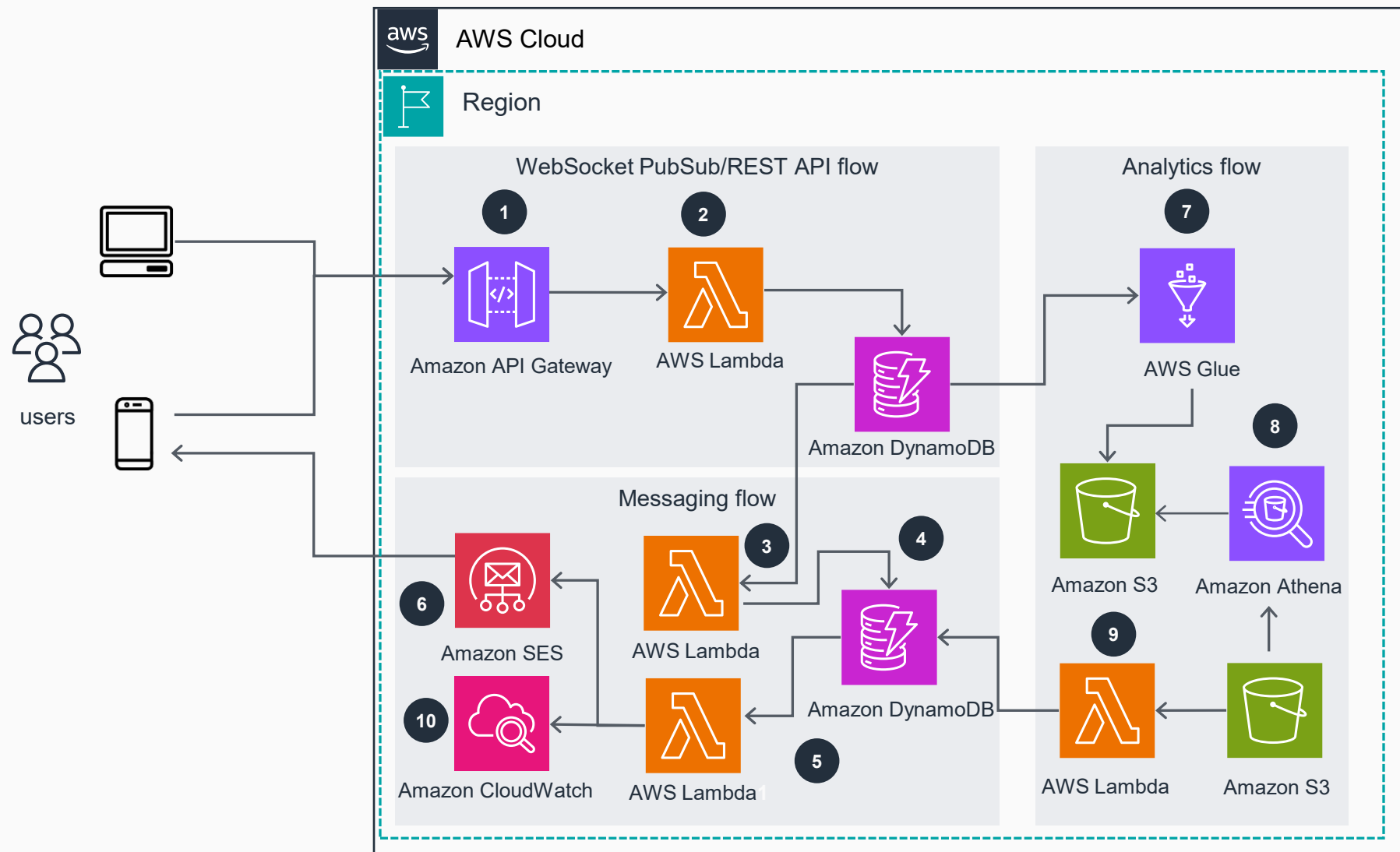


- 1 First, you'll connect through **Amazon API Gateway** WebSocket - this triggers an **AWS Lambda** function to handle your game events.
- 2 When you need to handle player data, a **Lambda** function takes your game and user information and stores it safely in an **Amazon DynamoDB** table.
- 3 You'll see your data being processed automatically by **DynamoDB Streams** triggering **Lambda** as soon as it arrives. This means when your players do something important (like friend login), you can notify them right away.
- 4 To keep track of all your messages, you'll store them in a Message table in **Amazon DynamoDB**. This way, you can see what you've sent and check if it was delivered successfully.
- 5 When **DynamoDB Streams** detect new message content in your DynamoDB table, a **Lambda** function automatically prepares it for sending through **Amazon Simple Email Service (Amazon SES)**.
- 6 **Amazon SES** then handles delivering your notifications - making sure they reach your players' devices exactly when you want them to.
- 7 When you need to send messages to specific groups of players (like "all players active in last 30 days"), you'll use **DynamoDB Zero-ETL** and **AWS Glue** to organize your player data.
- 8 You can then run your custom queries in **Amazon Athena** to find exactly which players should get your messages and save your results to an **Amazon S3** bucket.



Guidance for Games Messaging System on AWS

This architecture shows a flexible gaming messaging system that delivers real-time notifications for friend activities and targeted marketing campaigns. This serverless architecture reduces costs and scales automatically.



- 9 A **Lambda** function will then take these results, add them to the Message table in **DynamoDB**, and start your notification process automatically.
- 10 Throughout all of this, you can monitor your system's performance through **Amazon CloudWatch** dashboards, seeing exactly how your messages are being delivered.

