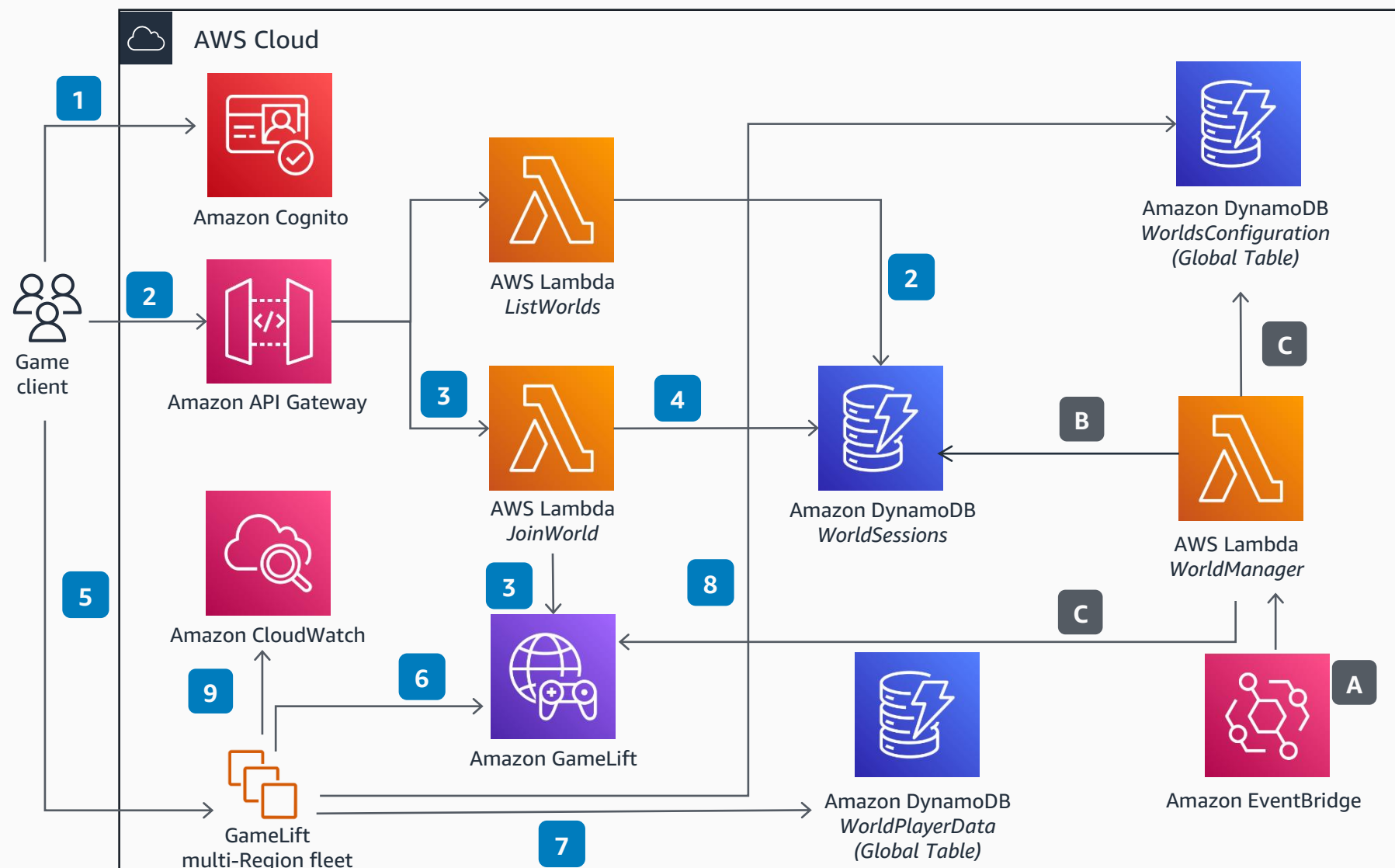


Guidance for Persistent World Game Hosting on AWS

This architecture hosts virtual worlds globally and persists world session and world data to a database. Steps A-C represent the back end of the system, and Steps 1-9 represent the front end.



Reviewed for technical accuracy February 28, 2022
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Reference Architecture

BACK END

FRONT END

- A** Amazon EventBridge triggers WorldManager AWS Lambda function every one minute. The function checks existing worlds' status through the Amazon GameLift API.
- B** WorldManager Lambda function stores the current state of the sessions and worlds to Amazon DynamoDB for faster back end access.
- C** WorldManager queries configured worlds from DynamoDB and creates any worlds that are not running by calling the Amazon GameLift API CreateGameSession.

- 1** Game client request an identity and credentials from Amazon Cognito Identity Pool to sign authorized API requests.
- 2** Game client requests the world list through Amazon API Gateway. API Gateway triggers ListWorlds Lambda function that checks game session information in the defined Region from DynamoDB.
- 3** Game client requests to join a specific world in a specific Region by calling JoinWorld Lambda function through API Gateway.
- 4** Lambda function creates a player session for the player, increases the amount of players for that world in DynamoDB, and sends the connection information to the game client.
- 5** Client connects to the Amazon GameLift session directly over transmission control protocol (TCP) and sends the player session ID.
- 6** Amazon GameLift session validates the player session ID with the Amazon GameLift server software development kit (SDK).
- 7** Amazon GameLift checks for world-specific player data and updates the player data as needed. It will store the latest player location after the player leaves.
- 8** Amazon GameLift session checks DynamoDB for scheduled termination and terminates if requested.
- 9** Game server sends logs and metrics to Amazon CloudWatch using the CloudWatch Agent.