



Architecture Diagrams Game Tech

Serverless Notifications for Mobile Games



Serverless Notifications for Mobile Games: Architecture Diagrams

Game Tech

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Notifications with Step Functions 1

Serverless Notifications with Step Functions diagram 1

Further reading 2

Diagram history 2

Notifications with Pinpoint 4

Serverless Notifications with Amazon Pinpoint diagram 4

Further reading 5

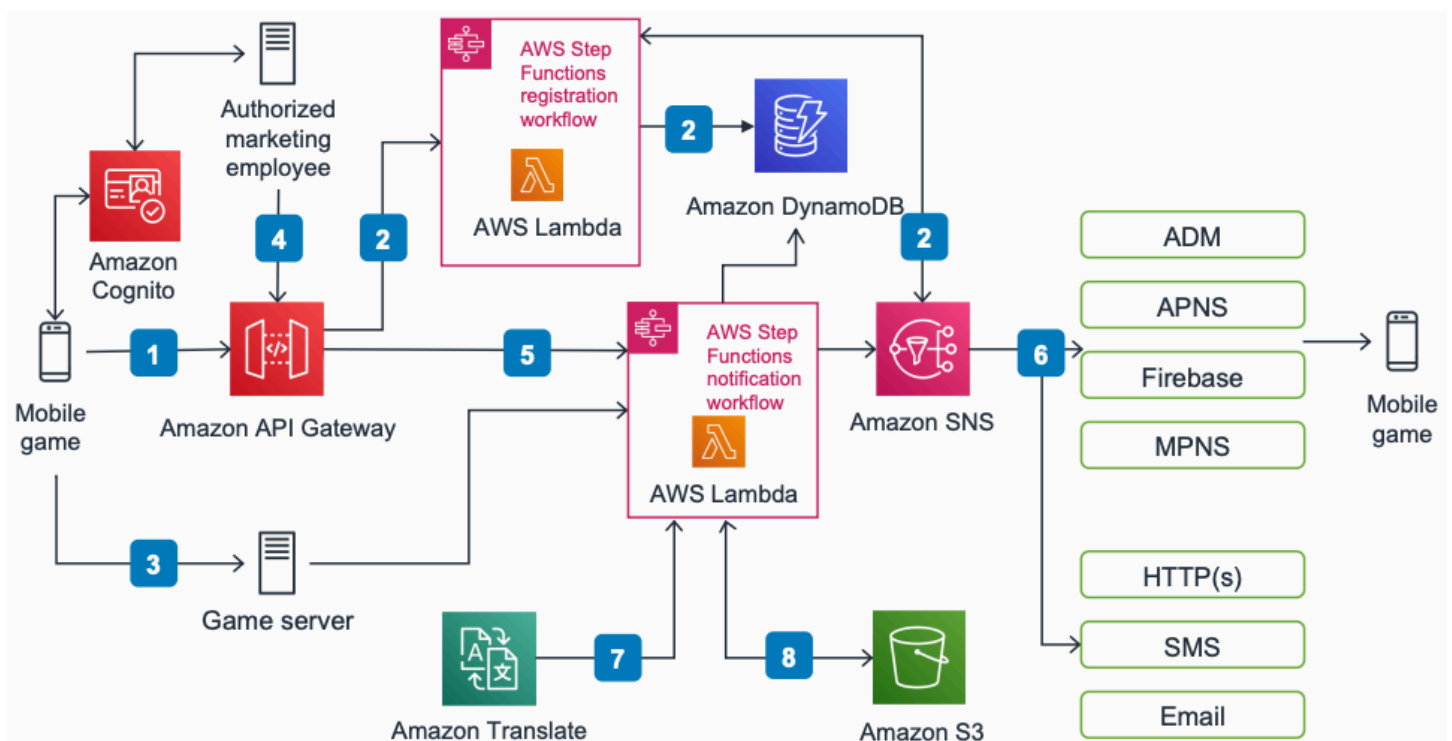
Diagram history 5

Serverless Notifications for Mobile Games with Step Functions

Publication date: **March 10, 2021** ([Diagram history](#))

This architecture creates a serverless data flow to ingest, store, process, and send push notifications to subscribers. Use this pipeline for flash offers, in-game event notifications, and campaign-specific notifications targeting segmented audiences with efficiency tracking.

Serverless Notifications with Step Functions diagram



The following steps describe the architecture:

1. When the game client launches, it sends registration information to an [Amazon API Gateway](#) REST endpoint. The game client can also send direct notification information through REST.
2. API Gateway requests the [AWS Step Functions](#) registration flow that performs additional logic like device registration and storing [Amazon Simple Notification Service](#) endpoint data in [Amazon DynamoDB](#).
3. As the game client sends information to the game server, the server triggers the Step Functions notification workflow that processes messages.

4. To run campaigns or send single notifications, a marketing employee can also manually request the Lambda Step Functions flow through API Gateway. Authorization is handled through [Amazon Cognito](#).
5. The notification flow performs actions like translation (by using [Amazon Translate](#)), message payload enrichment, and storing and obtaining endpoint data from DynamoDB. This data defines direct users or segments to send messages to.
6. Step Functions uses Amazon Translate for quick translation of notification content.
7. Amazon SNS uses the push notification service credentials and mobile device tokens. After receiving the message, all topic subscribers receive the push notification through providers like Firebase or Apple push notification service.
8. Raw data is sent to [Amazon Simple Storage Service](#) for cold data analytics.

Further reading

For additional information, see the following resources:

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	March 10, 2021
Initial publication	Reference architecture diagram first published.	March 10, 2021

 Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.

Serverless Notifications for Mobile Games with Amazon Pinpoint

Publication date: **March 10, 2021** ([Diagram history](#))

This architecture provides an alternative notification flow that uses [Amazon Pinpoint](#) for campaign-based messaging, A/B testing, and segmented audience targeting with built-in streaming analytics.

Serverless Notifications with Amazon Pinpoint diagram



The following steps describe the architecture:

1. The game client sends notification information to an [Amazon API Gateway](#) REST endpoint. You can update the API version and scale to handle large numbers of requests automatically.
2. To access underlying AWS services through the SDK, the mobile game client requests temporary credentials from [Amazon Cognito](#) Identity pools.
3. The mobile game client can also use the AWS Mobile SDK to invoke Amazon Pinpoint directly with a payload. The device can register or update endpoints dynamically.
4. API Gateway requests Amazon Pinpoint, which performs additional logic like message token translation, segmented message delivery, and streaming analytics for further analysis.

5. [Amazon Simple Notification Service](#) uses the push notification service credentials and mobile device tokens. After receiving the message, all subscribers receive the push notification through providers like Firebase or Apple push notification service.
6. An [AWS Lambda](#) function subscribes to [DynamoDB](#) Streams to sync users to Amazon Pinpoint.
7. A marketing employee can initiate a campaign, personal message, or A/B testing through the AWS console for Amazon Pinpoint.
8. Campaign and message sending data is streamed to Amazon Kinesis for streaming event analytics or to be stored for later insights in a data lake.

Further reading

For additional information, see the following resources:

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	March 10, 2021
Initial publication	Reference architecture diagram first published.	March 10, 2021

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.