



Architecture Diagrams Game Tech

Dynamic Media Content for Games



Dynamic Media Content for 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

Dynamic media content **i**

Dynamic Media Content for Games diagram 1

Further reading 2

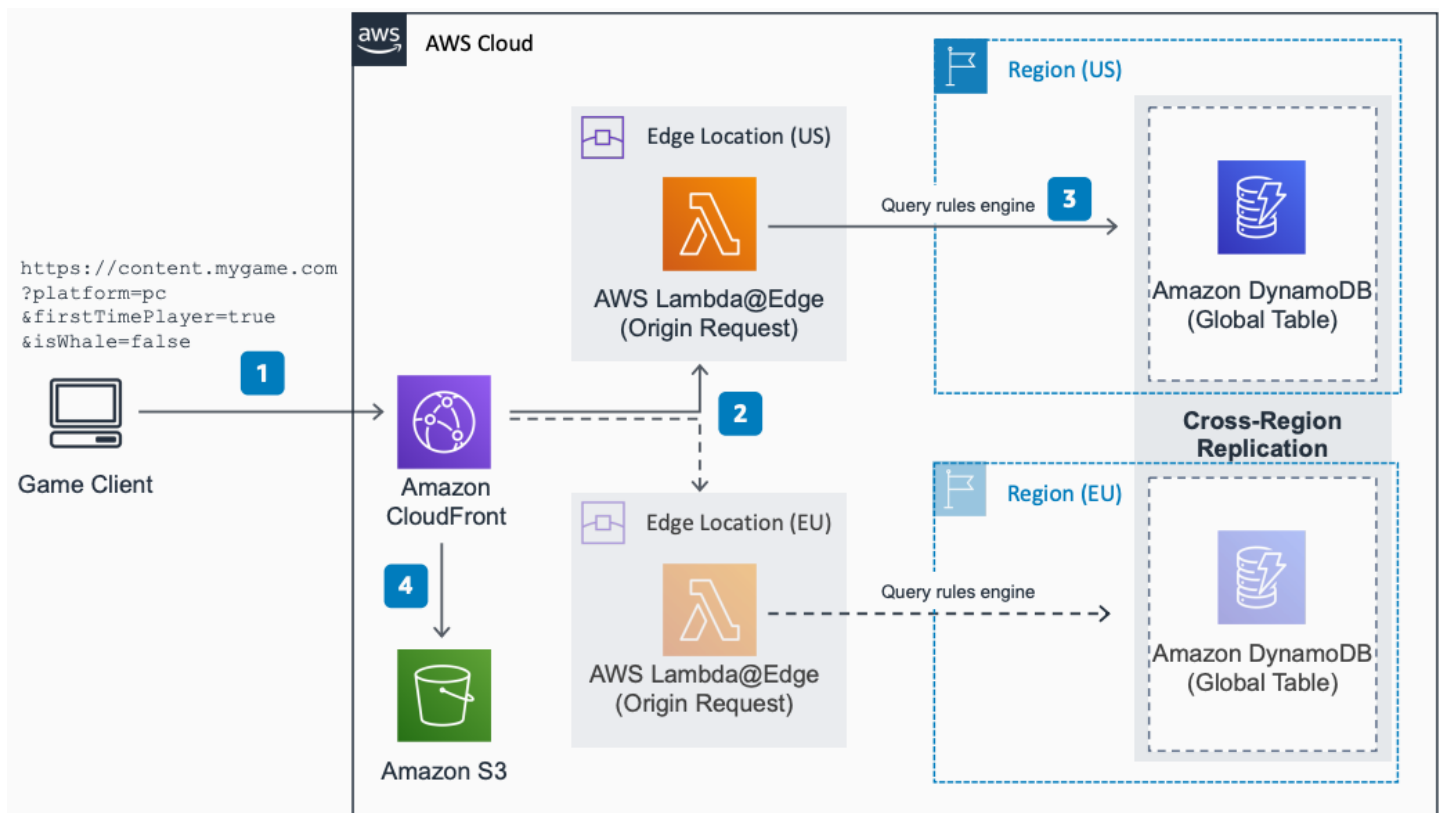
Diagram history 2

Dynamic Media Content for Games

Publication date: **October 8, 2020** ([Diagram history](#))

Game developers often build custom solutions to engage with players through daily messages and custom media content, including Message of the Day (MOTD). This architecture provides a serverless approach for serving customized content based on request parameters and edge computing.

Dynamic Media Content for Games diagram



The following steps describe the architecture:

1. Game clients send a request with parameters to [Amazon CloudFront](#) to download dynamic content. Request parameters represent attributes about the player, but use low-cardinality values to maintain high cache rates and reduce cost.
2. CloudFront invokes an AWS [Lambda@Edge](#) function closest to the user. The function dynamically modifies the content that should be retrieved from origin based on request parameters.

3. [Amazon DynamoDB](#) Global Tables provides a multi-Region rules engine for custom routing. The Lambda function queries the table closest to the user to determine the content to serve.
4. CloudFront uses the response from Lambda to fetch the relevant content from [Amazon Simple Storage Service](#) (origin). CloudFront returns the content to the client and caches the response in edge locations to reduce cost and improve performance for future requests.

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.	October 8, 2020

Note

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