



Architecture Diagrams Game Tech

Unreal Engine's Pixel Streaming on AWS



Unreal Engine's Pixel Streaming on AWS: 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

Pixel streaming **i**

Unreal Engine's Pixel Streaming on AWS diagram 1

Further reading 2

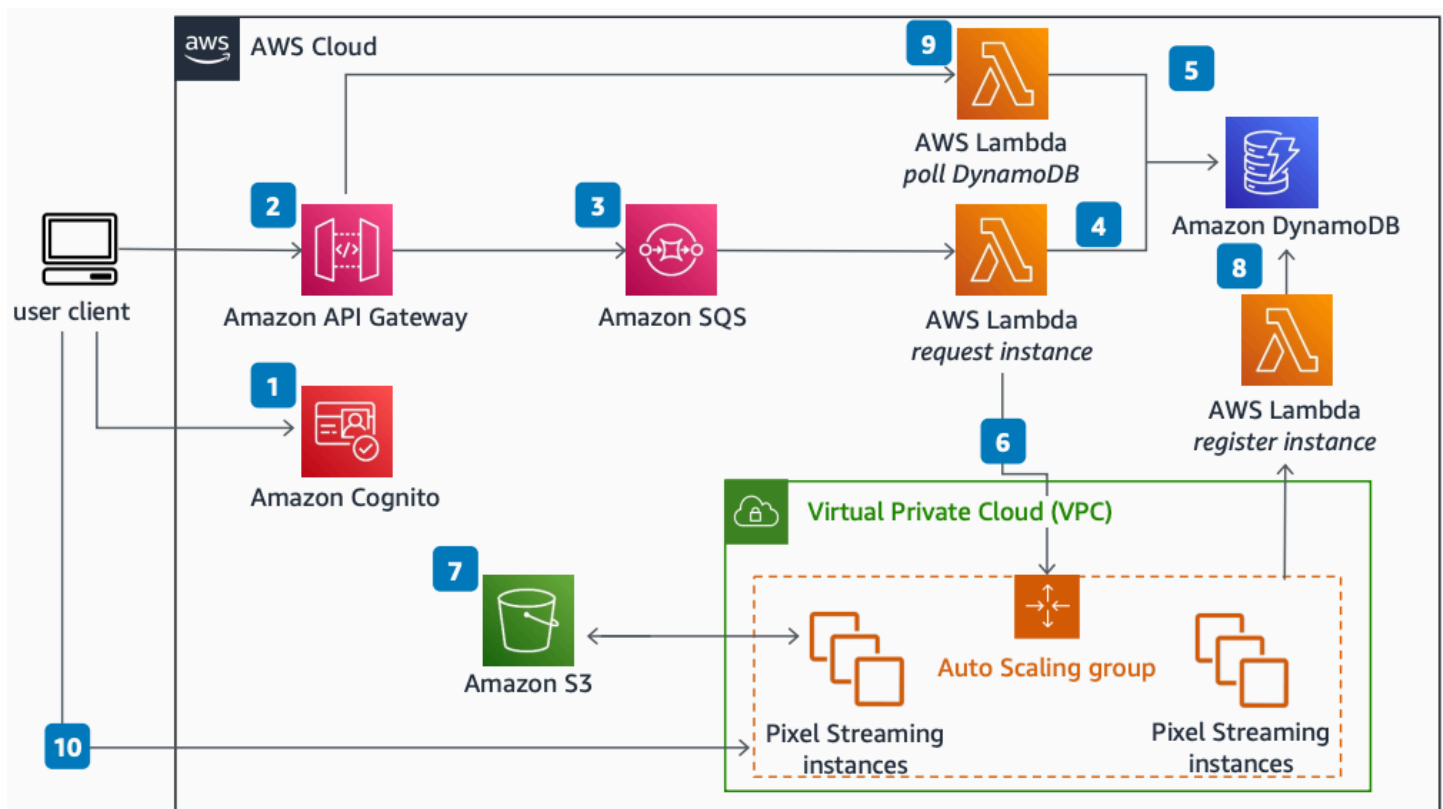
Diagram history 2

Unreal Engine's Pixel Streaming on AWS

Publication date: **January 6, 2023** ([Diagram history](#))

This architecture shows a simple implementation of Unreal Engine's Pixel Streaming feature on AWS. The architecture uses a serverless matchmaker composed of [Amazon Simple Queue Service](#), [AWS Lambda](#), and [Amazon DynamoDB](#). Both the streaming instance and signaling servers are hosted in the same [Amazon Elastic Compute Cloud](#) instance.

Unreal Engine's Pixel Streaming on AWS diagram



The following steps describe the architecture:

1. The client requests an [Amazon Cognito](#) identity and temporary AWS credentials.
2. The client signs a request to [Amazon API Gateway](#) with the temporary credentials.
3. API Gateway forwards the request to Amazon SQS to ensure requests are handled in order.
4. Lambda checks DynamoDB for an available streaming instance. If none are available, it can request a new instance.

5. DynamoDB stores required metadata such as the instance status, IP address, port number, and user reservation details.
6. If an instance is not available, Lambda requests that a new one be created.
7. [Amazon Simple Storage Service](#) stores static assets such as game builds and bootstrap scripts.
8. After a new instance is created, its metadata is written to DynamoDB by using Lambda.
9. A separate route in API Gateway lets the client poll DynamoDB for their assigned instance.
10. The client uses the IP and port returned from DynamoDB to connect to their assigned instance.

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.	January 6, 2023

Note

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