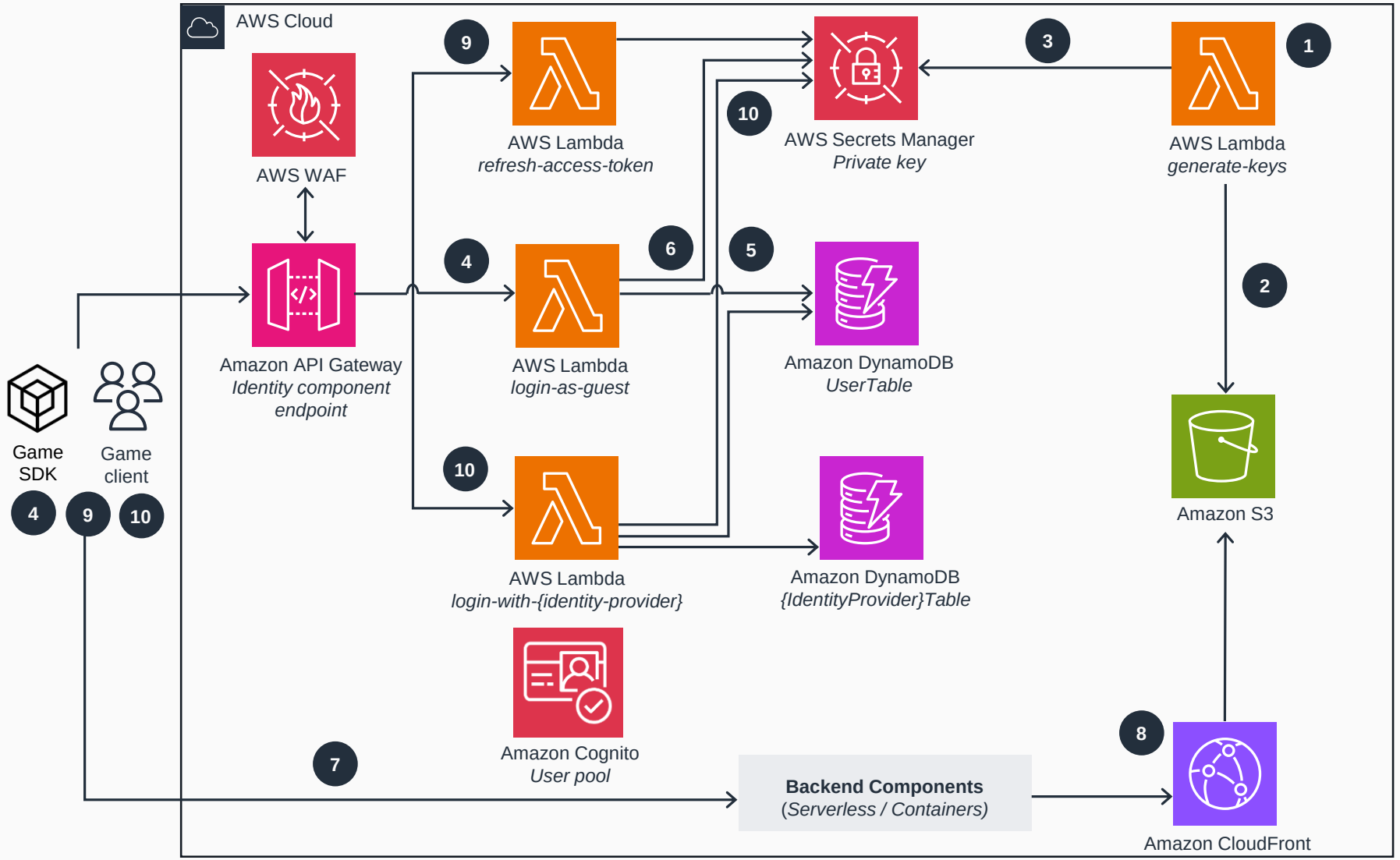
Guidance for Custom Game Backend Hosting on AWS

This architecture diagram illustrates how to deploy a custom, lightweight, and scalable cross-platform game identity component and how to use the identities to authenticate against custom game backend components on AWS. (Steps 1-9)



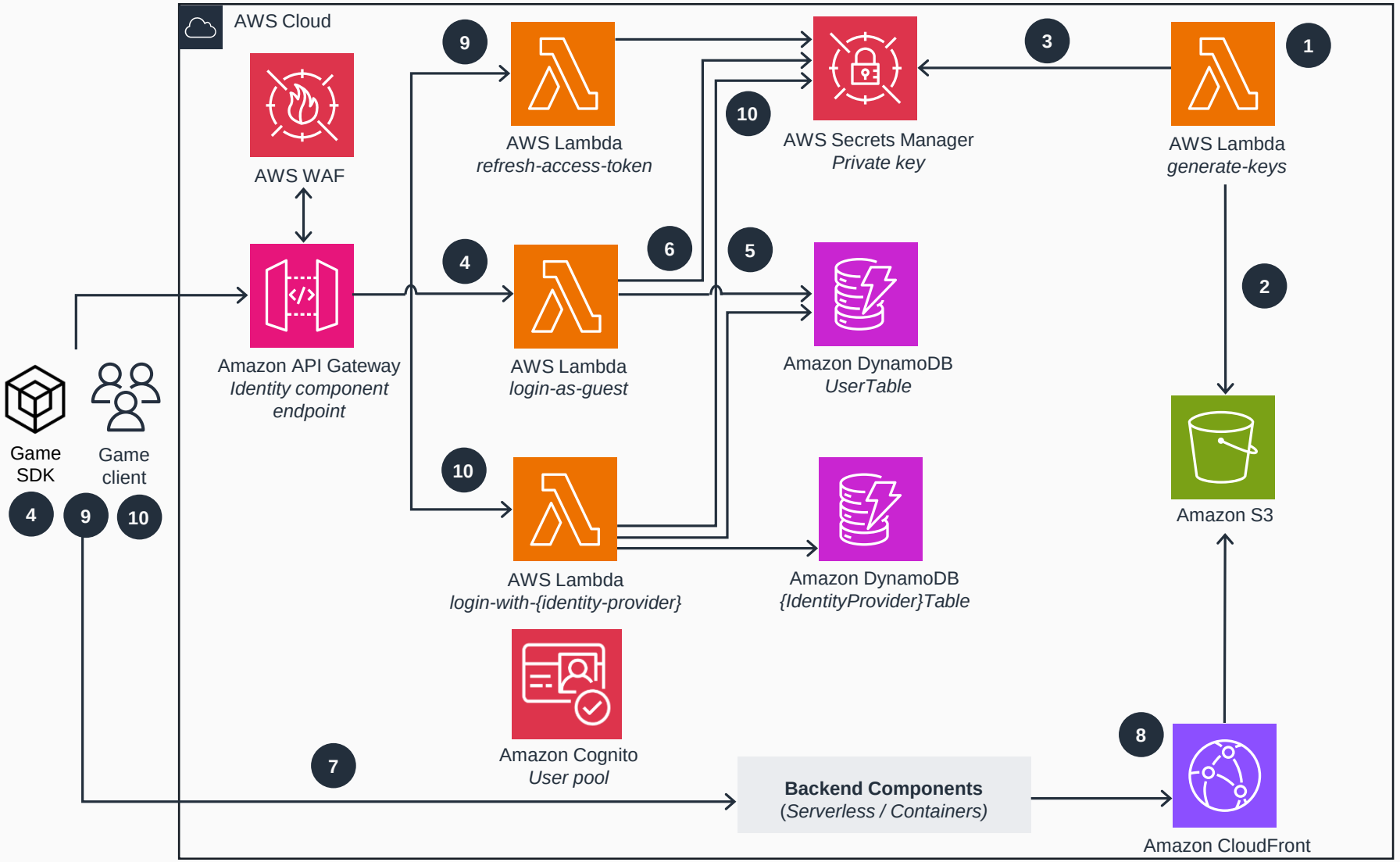
1. The **AWS Lambda** function *generate-keys* is invoked every 7 days.
2. *Generate-keys* gets the latest public *jwks.json* file from **Amazon Simple Storage Service** (Amazon 3), generates new public keys (JSON Web Key Set (JWKS)), and private keys, and updates **Amazon S3** with the new public key and the previous key.
3. *Generate-keys* updates the private key that is used to generate JSON Web Tokens (JWT) to **AWS Secrets Manager**.
4. The game client uses the software development kit (SDK) to request a new guest identity. Or, the game client can sign in with their existing guest identity by sending the *guest_secret* through **Amazon API Gateway**, which is protected by **AWS WAF** rules.
5. The *login-as-guest* **Lambda** function validates the guest identity, or creates a new one to the *UserTable* in **Amazon DynamoDB**.
6. The **Lambda** function requests the private key from **Secrets Manager**, generates a signed *JWT token* for the client, and sends it back.
7. The game client can now call custom backend components by sending requests with the *JWT token* in the *Authorization* header using the SDK.
8. The backend components validate the token by requesting the JWKS public keys from the public endpoint through **Amazon CloudFront**, which gets the file from **Amazon S3**.
9. The SDK automatically refreshes the JWT access token by calling *refresh-access-token* **Lambda** function through **API Gateway.** The function generates a new token using the private key from **Secrets Manager.**

Reviewed for technical accuracy February 26, 2024

AWS Reference Architecture

Guidance for Custom Game Backend Hosting on AWS

This architecture diagram illustrates how to deploy a custom, lightweight, and scalable cross-platform game identity component and how to use the identities to authenticate against custom game backend components on AWS. (Step 10)



10 Additionally, the game client can send access tokens from the game platform-specific identity provider to link to an existing account, or create a new account. This account can optionally be created in an **Amazon Cognito** user pool. The **Lambda** functions validate the tokens and create the link to the user account in a specific **DynamoDB** table. Then it generates a *JWT token* for the client using the private key from **Secrets Manager**.

Reviewed for technical accuracy February 26, 2024

AWS Reference Architecture