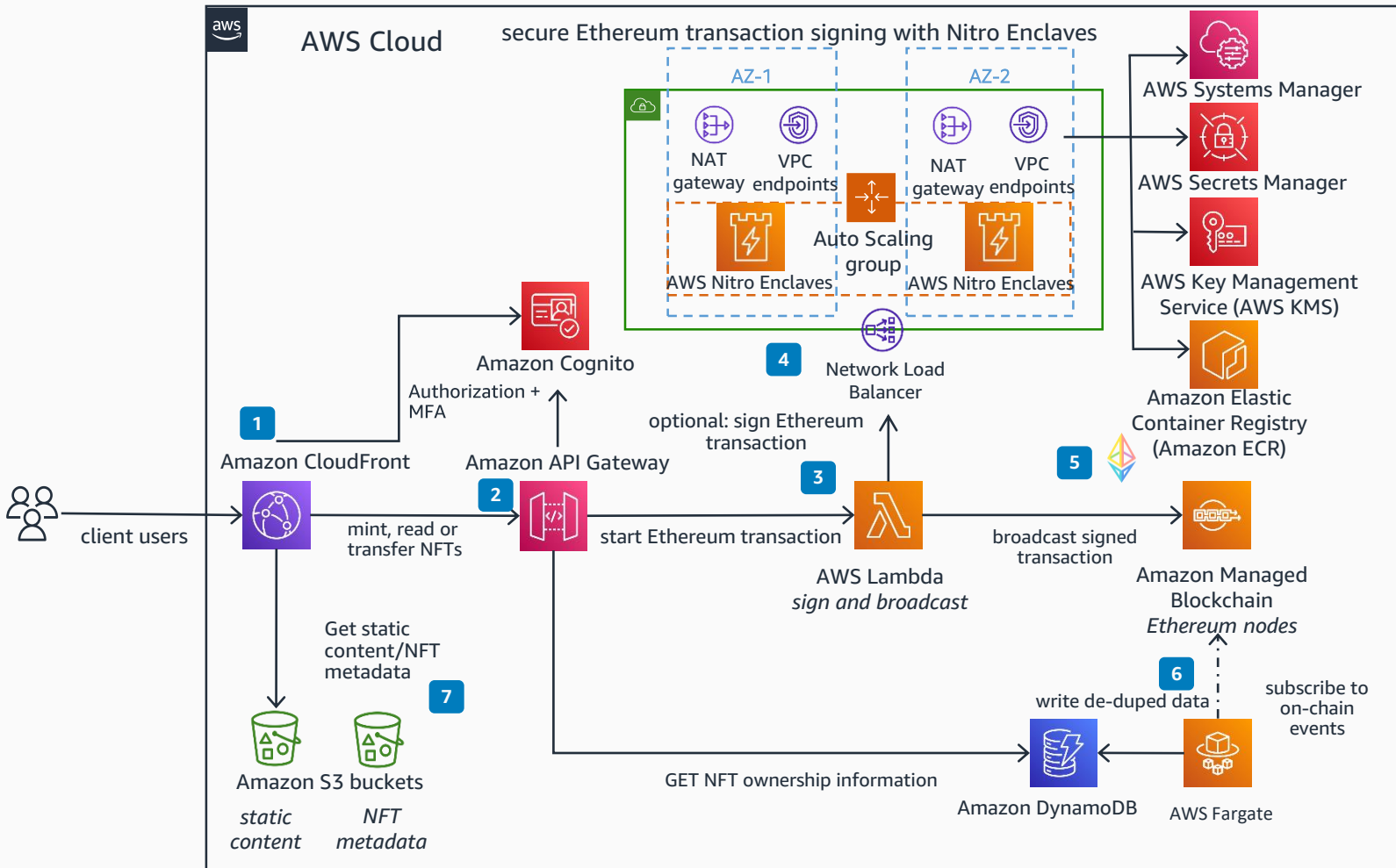


Guidance for Minting Ethereum-based NFTs on AWS

Facilitate non-fungible token operations on the Ethereum blockchain for your users

With AWS serverless technologies and Amazon Managed Blockchain, you can deploy a scalable platform on which to mint and manage NFTs for a broad set of users in a custodial environment.



- 1 The web client user authenticates with the **Amazon Cognito** User Pool, requesting a JSON Web Token (JWT) that will be verified upon requests to the REST API.
- 2 The web client user submits a POST request to mint a non-fungible token (NFT), providing both the required arguments for the request and the JSON web token (JWT) to authorize the operation to the **Amazon API Gateway** endpoint. **Amazon API Gateway** integrates with **Amazon Cognito** to verify the JWT and its contents.
- 3 After authorization, the **Amazon API Gateway** request cues an **AWS Lambda** function that's responsible for building and signing the Ethereum transaction for the mint operation.
- 4 Using the user's unique identifier encoded in the JWT provided for the request, the **Lambda** function triggers the secure transaction signing module that utilizes isolated compute instances in **AWS Nitro Enclaves** to sign the Ethereum transaction with the user's highly sensitive private key in custody. In the transaction signing module, **AWS Systems Manager** manages access to the **Amazon Elastic Compute Cloud** (Amazon EC2) instance(s), **AWS Key Management Service** (AWS KMS) manages the symmetric encryption key used to derive private keys, and **AWS Secrets Manager** securely manages the encrypted private keys (ciphertext).*
- 5 Once the transaction is securely signed with the user's private key, the now signed transaction is broadcast to the public **Ethereum** network via the JSON-RPC API exposed by a fully managed Ethereum full node on **Amazon Managed Blockchain**. The Ethereum node returns a transaction hash (ID) that can be used to retrieve information about the transaction and its status on the blockchain. In addition, the **Lambda** function stores any associated JSON metadata for the NFT in an **Amazon Simple Storage Service** (Amazon S3) bucket.
- 6 To monitor the blockchain transaction's status and its resulting state mutation on the network, an **AWS Fargate** task manages an **Amazon Elastic Container Service** (Amazon ECS) application that subscribes to on-chain events and writes updates to a noSQL database, **Amazon DynamoDB**, for later query when a relevant event is published. For example, subscribing to events related to a transaction hash that will cue when a transaction is mined (included into a block) on the blockchain.
- 7 Once the transaction is included in a block and state is updated on the blockchain, the web client user can query the ownership of their new NFT and retrieve its metadata/content from the **Amazon S3** bucket.

*This example refers to a *custodial* key management environment (backend managed cryptographic keys), however, users in a self-custody environment (self-managed cryptographic keys) can sign transaction(s) directly from the web client interface using their own wallet, bypassing the backend signing using Nitro Enclaves.