Developing Secure Applications on AWS

SHARED RESPONSIBILITY · TLS/SSL · ACM CERTIFICATES · AWS STS · AUTHN VS AUTHZ · AMAZON COGNITO · USER & IDENTITY POOLS · JWT TOKENS

STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

01 **Two jobs, one model:** under the **shared responsibility model** AWS secures *of* the cloud and you secure *in* the cloud — at **every layer** of the app. This module covers the two developer-facing aspects: **securing network connections** between clients and application servers over the internet, and **controlling user access** to the AWS resources your application calls.

02 **Encrypt in transit: SSL/TLS** protocols encrypt communications and use a **certificate issued by a certificate authority (CA)** to establish a website's identity. **ELB** terminates HTTPS/TLS at the load balancer; **CloudFront** negotiates HTTPS to viewers and origin. **AWS Certificate Manager (ACM)** provisions, deploys, and **auto-renews** public and private certs — *free* for AWS services (ELB, CloudFront, API Gateway); private certs via **ACM Private CA**.

03 **Temporary credentials are the rule:** never embed access keys in code or share them. Use **IAM roles**; if roles aren't an option, use **AWS STS**. STS credentials = **access key ID + secret access key + session token**, short-lived with a configurable lifetime, **generated dynamically**, and **not reusable after they expire**.

04 **AuthN ≠ AuthZ: authentication** verifies *who* the user is; **authorization** verifies *what* they may do. An **identity provider (IdP)** manages identities; an **identity broker** authenticates credentials against the IdP and fetches STS credentials. **SAML** (enterprise SSO), **OIDC** (third-party/social), and **JWT** (secure token transport) are the open standards.

05 **Cognito = two pools: user pools** are user directories for sign-up/sign-in (authentication) and return **three JWTs** — *ID* and *access* tokens (**signed**) plus a *refresh* token (**encrypted**). **Identity pools** (federated identities) trade an identity for **temporary AWS credentials** via an STS role (authorization to reach AWS services). Use them **separately or together**.

02 KEY TERMS

TERM	DEFINITION
TLS / SSL	Transport Layer Security and its predecessor Secure Sockets Layer — open standards that use certificates to establish website identity and encrypt communications between connected resources
Certificate authority (CA)	Trusted issuer of certificates to specific domains; a certificate from a trusted CA tells a browser or app the connection is safe
AWS Certificate Manager (ACM)	Provisions, manages, deploys, and renews public and private SSL/TLS certificates for AWS services and internal resources; certs for AWS services are free
Authentication	Verifying the identity of the user (who they are)
Authorization	Verifying the user's permissions (what the user is allowed to do)
Identity provider (IdP)	System that manages identity information and provides authentication services
Identity broker	Software layer that authenticates credentials against an IdP and retrieves temporary security credentials from AWS STS
AWS STS	AWS Security Token Service — issues short-lived credentials (access key ID, secret access key, session token) to trusted users
Federated identity	A user who signs in from an authentication system outside AWS — enterprise (SSO via SAML 2.0) or web (social sign-in via a third-party OIDC IdP)
JSON Web Token (JWT)	Open standard for securely transmitting information between two parties; the token format Amazon Cognito user pools issue
User pool	Amazon Cognito user directory that provides sign-up/sign-in and returns ID, access, and refresh tokens

TERM	DEFINITION
Identity pool	Amazon Cognito federated identities that provide temporary, limited-privilege AWS credentials through an STS role

03 AMAZON COGNITO — USER POOLS VS. IDENTITY POOLS

who you are vs. what you can reach

USER POOLS

Authentication — verify who the user is

- Directory of users; sign-up and sign-in for your app
- Accept federated identities from Login with Amazon, Google, Facebook, or any SAML/OIDC IdP
- Return **three JWTs** (ID, access, refresh) on successful sign-in
- Built-in customizable web UI; compromised-credentials check; phone/email verification; adaptive authentication

IDENTITY POOLS (FEDERATED IDENTITIES)

Authorization — reach AWS resources

- Provide **temporary, limited-privilege AWS credentials**
- Trade an identity for credentials via an **AWS STS** role
- Create unique identities and federate them with many providers — a user pool is just one more IdP
- Support Cognito user pools, public/SAML/OIDC IdPs, and developer-authenticated identities

04 AWS STS API OPERATIONS

match the caller to the operation

OPERATION	RETURNS TEMPORARY CREDENTIALS TO…
`AssumeRole`	Existing **IAM users** for **cross-account** access to AWS resources
`AssumeRoleWithSAML`	**Federated** users authenticated by an organization's existing (SAML 2.0) identity system
`AssumeRoleWithWebIdentity`	**Federated** users authenticated through a public IdP (Login with Amazon, Facebook, Google, OIDC) — but AWS recommends using **Amazon Cognito** instead
`GetFederationToken`	Federated users; default expiration is longer — **12 hours** instead of 1 hour
`GetSessionToken`	Existing IAM users for enhanced security — e.g., allowing AWS requests only when **MFA** is enabled

05 THE THREE USER POOL TOKENS

signed = readable + verifiable · encrypted = opaque

TOKEN	WHAT IT DOES	PROTECTION
ID token	Authorizes API calls; carries claims about the user's identity (for example, name and email)	**Signed**
Access token	Authorizes API calls based on the custom (OAuth) scopes of access-protected resources	**Signed**
Refresh token	Holds the information needed to obtain a new ID or access token	**Encrypted**

06 USER POOL SIGN-IN FLOW

01 Sign in through the **user pool** or a third-party IdP → **02** **Authenticate** and get three JWTs from Amazon Cognito → **03**

Use tokens to get **AWS credentials** or authorize calls at **API Gateway**

07 EXAM TRAPS

where the knowledge check gets you

- × “SSL and TLS are rival protocols you must choose between.” — TLS is the **successor** to SSL; HTTPS may use either, but TLS is what typically secures the connection today. The term SSL persists out of habit.
- × “You must buy and manually renew your certificates.” — ACM **provisions and auto-renews** them, and public *and* private certificates provisioned through ACM for use with AWS services are **free** — you pay only for the AWS resources you run.
- × “Authentication and authorization are the same check.” — **Authentication** verifies *identity* (who you are); **authorization** verifies *permissions* (what you're allowed to do). Cognito user pools do the first, identity pools enable the second.
- × “A long-lived access key in code is fine if the repo is private.” — Embedding keys in unencrypted code and sharing keys between users are the two named **bad practices**. Use IAM roles, or AWS STS **temporary** credentials.
- × “For mobile/web social sign-in, call `AssumeRoleWithWebIdentity` directly.” — AWS **recommends Amazon Cognito** (and its credentials provider with the AWS SDKs) instead of calling that API directly.
- × “All three Cognito tokens are encrypted.” — The **ID and access tokens are signed, not encrypted**; only the **refresh token is encrypted**. Signed tokens are readable but tamper-evident.
- × “User pools and identity pools do the same job.” — A **user pool** is a directory that authenticates users and issues tokens; an **identity pool** issues **temporary AWS credentials**. They can be used separately or together.
- × “Track federated user actions by their IAM user name.” — Federated users have no IAM identity; **CloudTrail** records the `AssumeRoleWithSAML` and `AssumeRoleWithWebIdentity` calls and the assumed **role ARN**, and the user appears in the `userName` attribute.

08 SELF-QUIZ

answers are upside-down below

Q1 Name the two aspects of application security this module covers.

Q2 Which AWS service provisions, deploys, and renews SSL/TLS certificates and integrates with ELB, CloudFront, and API Gateway?

Q3 What is Elastic Load Balancing's role in TLS for a web application?

Q4 What three values make up a set of AWS STS temporary security credentials?

Q5 State the difference between authentication and authorization.

Q6 What is the job of an identity broker?

Q7 Which STS API operation returns credentials to federated users authenticated by a SAML 2.0 identity system?

Q8 Which two STS API calls does CloudTrail record to let you track federated user actions?

Q9 In Amazon Cognito, which component is a user directory for sign-in, and which provides temporary AWS credentials?

Q10 Of the three user pool tokens, which one is encrypted rather than only signed, and what is it used for?

ANSWER KEY (rotate the page)

Q1 Securing network connections between clients and application servers, and controlling user access to the AWS resources the application calls **Q2** AWS Certificate Manager (ACM) **Q3** It terminates HTTPS/TLS traffic at the load balancer (encrypting/decrypting) so each EC2 instance does not have to handle TLS termination **Q4** An access key ID, a secret access key, and a session token **Q5** Authentication verifies the user's identity (who); authorization verifies the user's permissions (what they are allowed to do) **Q6** It authenticates credentials against an IdP and retrieves temporary security credentials from AWS STS **Q7** AssumeRoleWithSAML **Q8** AssumeRoleWithSAML and AssumeRoleWithWebIdentity **Q9** The user pool is the directory for sign-in; the identity pool provides temporary AWS credentials **Q10** The refresh token; it holds the information needed to obtain a new ID or access token