[ENGLISH]

Hello Cloud Gurus

and welcome to this lecture.

And this lecture is

the Advanced Identity Access Management Summary.

So, it's the end of this section of the course

and it's going to be all summary

and exam tips for Advanced Identity Access Management.

So, let's get started.

Remember, Web Identity Federation allows users

to authenticate with a web identity provider,

so think Google, Facebook or Amazon.

The user authenticates first with the web ID provider,

they then receive an authentication token

which is then exchanged for temporary AWS credentials

allowing them to assume an Identity Access Management role

and the IAM role allows them to access AWS resources.

Remember the STS assume role with web identity API call.

And this is part of the SDS or the security token service

and it allows users who have authenticated

with a web identity provider to access AWS resources.

And remember that after the user

has authenticated successfully,

at that point the application makes the assume role

with web identity API call.

And if successful, STS will return temporary credentials

enabling access to AWS resources.

And in the response from SDS, within AssumedRoleUser,

the ARN and the AssumedRole ID are used

to programmatically reference the temporary credentials

so the access key and secret access key returned

by STS and not an Identity Access Management role or user.

So don't get the two confused.

Cognito is an ID broker and it handles the interaction

between your applications and the web ID provider,

so you don't need to write your own code to do this.

It provides sign up, sign in and guest user access

and syncs user data for a seamless experience

across multiple devices.

And Cognito is the AWS recommended approach

for web ID Federation.

So, if you see any exam questions asking you

which technology to use

if you want to authenticate your applications

against a web identity provider like Facebook, Google

or Amazon, if there's any questions around that asking you

which is the best technology,

well Cognito is the recommended approach,

particularly for mobile applications.

And it uses push synchronization to send

a silent push notification via SNS,

Simple Notification Service,

of user updates to multiple device types.

So your mobile phone, tablet and laptop, for example,

all associated with your user ID.

So in our example, we had our user

authenticating against a user pool

which then receives JWT tokens,

so JSON web tokens from the web identity provider,

so Facebook, Google or Amazon.

The Cognito identity pool then exchanges the JWT tokens

for AWS credentials which enables the user

to temporarily access AWS resources.

For example, S3, DynamoDB, RDS, Easy2, et cetera.

And if you're struggling to remember

the difference between user pools and identity pools,

then just remember that user pools

are like user directories and they're used to manage signup

and sign in functionality for mobile and web applications.

Whereas identity pools enable you

to provide temporary AWS credentials,

so temporary credentials to enable access

to AWS services like S3 or DynamoDB.

So, essentially user pools are all about sign up

and sign in functionality.

Whereas it's the identity pools which give you

the mechanism to provide temporary credentials,

so the temporary access key and secret access key

to allow your users access to AWS services.

Moving on to inline policies,

managed policies and custom policies.

Remember the three different types of IAM policy.

So, we have the managed policy

and this is the AWS managed default policies.

So, the ones that we can search for

when we go to create a role or when we want to assign

a policy to a user group or role,

and these are the policies like DynamoDBFullAccess,

CodeDeployFullAccess, S3FullAccess, et cetera.

And these can be assigned to any number of users,

groups or roles across AWS.

The customer managed policy,

so that's for any policy that is not catered

for by the managed policy,

you can create your own and it's managed by you

and it can be attached to multiple users, groups, or roles.

And then finally, there's the inline policy

and that is managed by you

and embedded within a single user, group or role.

So, it's a one-to-one mapping

between the policy and the entity to which it's embedded

and when you delete the user, group or role

the policy is also deleted.

And in most cases, AWS recommends

using managed policies over inline policies.

And finally, remember how to configure

across account access at a high level.

And if you remember in our demo, we did this together.

So, we started off with our development account

and our production account.

We created our developers Identity Access Management group

and we also created our user called Anna,

and she was a member of the developers group.

Then in our production accounts, we created an S3 bucket

and an Identity Access Management policy

with permissions to list, and put,

and delete items from the S3 bucket.

We then attach that policy

to an Identity Access Management role

and the role that we created specified the account ID

of all development accounts.

And then finally, we created

an Identity Access Management policy

which enabled our developers group

to assume the role we created in production.

So, that is the end of this lecture

and that's the end of this section.

So, well done for completing the section.

If you have any questions please let me know,

if not, feel free to move on to the next lecture.

Thank you.

[SPANISH]

Hola gurús de la nube

y bienvenidos a esta conferencia.

Y esta conferencia es

el resumen de gestión avanzada de acceso a la identidad.

Entonces, es el final de esta sección del curso.

y va a ser todo resumen

y consejos para el examen de Gestión avanzada de acceso a la identidad.

Entonces empecemos.

Recuerde, Web Identity Federation permite a los usuarios

para autenticarse con un proveedor de identidad web,

así que piensa en Google, Facebook o Amazon.

El usuario se autentica primero con el proveedor de identificación web,

reciben un token de autenticación

que luego se intercambia por credenciales temporales de AWS

permitiéndoles asumir un rol de gestión de acceso a la identidad

y el rol de IAM les permite acceder a los recursos de AWS.

Recuerde que el STS asume el rol con la llamada a la API de identidad web.

Y esto es parte del SDS o del servicio de token de seguridad.

y permite a los usuarios que se han autenticado

con un proveedor de identidad web para acceder a los recursos de AWS.

Y recuerda que después del usuario

se ha autenticado con éxito,

en ese momento, la aplicación asume el rol

con llamada a la API de identidad web.

Y si tiene éxito, STS devolverá credenciales temporales

permitir el acceso a los recursos de AWS.

Y en la respuesta de SDS, dentro de AssumedRoleUser ,

se utilizan el ARN y el ID de rol asumido

para hacer referencia programáticamente a las credenciales temporales

por lo que la clave de acceso y la clave de acceso secreta regresaron

por STS y no por un rol o usuario de Identity Access Management.

Así que no confundas a los dos.

Cognito es un intermediario de ID y maneja la interacción.

entre sus aplicaciones y el proveedor de identificación web,

por lo que no necesita escribir su propio código para hacer esto.

Proporciona registro, inicio de sesión y acceso de usuario invitado.

y sincroniza los datos del usuario para una experiencia perfecta

a través de múltiples dispositivos.

Y Cognito es el enfoque recomendado por AWS

para la federación de identificación web.

Por lo tanto, si ve alguna pregunta de examen que le pregunte

que tecnologia usar

si desea autenticar sus aplicaciones

contra un proveedor de identidad web como Facebook, Google

o Amazon, si hay alguna pregunta al respecto preguntándole

cual es la mejor tecnologia

Bueno Cognito es el enfoque recomendado,

especialmente para aplicaciones móviles.

Y utiliza la sincronización push para enviar

una notificación push silenciosa a través de SNS,

Servicio de notificación simple,

de actualizaciones de usuario para múltiples tipos de dispositivos.

Entonces, su teléfono móvil, tableta y computadora portátil, por ejemplo,

todos asociados con su ID de usuario.

Entonces, en nuestro ejemplo, teníamos nuestro usuario

autenticación contra un grupo de usuarios

que luego recibe tokens JWT,

entonces tokens web JSON del proveedor de identidad web,

entonces Facebook, Google o Amazon.

El grupo de identidades de Cognito luego intercambia los tokens JWT

para las credenciales de AWS, lo que permite al usuario

para acceder temporalmente a los recursos de AWS.

Por ejemplo, S3, DynamoDB , RDS, Easy2, etcétera.

Y si estás luchando por recordar

la diferencia entre grupos de usuarios y grupos de identidades,

entonces recuerda que los grupos de usuarios

son como directorios de usuarios y se usan para administrar el registro

y funcionalidad de inicio de sesión para aplicaciones móviles y web.

Mientras que los grupos de identidades le permiten

para proporcionar credenciales temporales de AWS,

entonces credenciales temporales para habilitar el acceso

a servicios de AWS como S3 o DynamoDB .

Entonces, esencialmente, los grupos de usuarios se tratan de registrarse

e iniciar sesión en la funcionalidad.

Mientras que son los grupos de identidad los que le dan

el mecanismo para proporcionar credenciales temporales,

entonces la clave de acceso temporal y la clave de acceso secreta

para permitir que sus usuarios accedan a los servicios de AWS.

Pasando a las políticas en línea,

administradas y políticas personalizadas.

Recuerde los tres tipos diferentes de políticas de IAM.

Entonces, tenemos la política administrada

y estas son las políticas predeterminadas administradas por AWS.

Entonces, los que podemos buscar

cuando vamos a crear un rol o cuando queremos asignar

una política a un grupo de usuarios o rol,

y estas son las políticas como DynamoDBFullAccess ,

CodeDeployFullAccess , S3FullAccess, etcétera.

Y estos se pueden asignar a cualquier número de usuarios,

grupos o roles en AWS.

La política administrada por el cliente,

así que eso es para cualquier póliza que no sea atendida

por la política administrada,

puedes crear el tuyo propio y lo administras tú

y se puede adjuntar a múltiples usuarios, grupos o roles.

Y finalmente, está la política en línea

y eso lo manejas tu

e incrustado dentro de un solo usuario, grupo o rol.

Entonces, es un mapeo uno a uno

entre la política y la entidad a la que está integrada

y cuando elimina el usuario, grupo o rol

la política también se elimina.

Y en la mayoría de los casos, AWS recomienda

usando políticas administradas sobre políticas en línea.

Y por último, recuerda cómo configurar

a través del acceso a la cuenta a un alto nivel.

Y si recuerdas en nuestra demostración, hicimos esto juntos.

Entonces, comenzamos con nuestra cuenta de desarrollo

y nuestra cuenta de producción.

Creamos el grupo de administración de acceso a la identidad de nuestros desarrolladores

y también creamos nuestro usuario llamado Anna,

y ella era miembro del grupo de desarrolladores.

Luego, en nuestras cuentas de producción, creamos un depósito S3

y una política de gestión de acceso a la identidad

con permisos para listar, y poner,

y elimine elementos del depósito S3.

Luego adjuntamos esa política

a un rol de administración de acceso a la identidad

y el rol que creamos especificó el ID de la cuenta

de todas las cuentas de desarrollo.

Y finalmente, creamos

una política de gestión de acceso a la identidad

que permitió a nuestro grupo de desarrolladores

para asumir el papel que creamos en la producción.

Entonces, ese es el final de esta conferencia.

y ese es el final de esta sección.

Bien hecho por completar la sección.

Si tiene alguna pregunta, por favor hágamelo saber,

si no, siéntase libre de pasar a la siguiente lección.

Gracias.