**Access keys**

Tools to access AWS access keys are the way in which the AWS command line tools and many other applications authenticate to AWS in the same way that you identify to the console. Using a username, improve your identity with a password and optionally an MFA access keys do the same for applications.

**Access keys is actually a single thing**, although the name suggests it's multiple things. The term access keys refers to a **pair of keys, a pair of keys that need to be used together.** First, there's the **access key ID. And that's the public part.** In many ways, it's similar to a user name. **It identifies the access key, and it's something that both you and AWS possess**. You can always get the access key ID. It's in the AWS console for any access keys which exist now. The second part of an access key pair is known as the **Secret Access Key.** The secret access key is like your password. It's generated. When you generate the access keys, you can note it down, store it somewhere, do whatever you want with it, but it's only available. **The ones when you make the set of access keys is the only time that you're able to get this from AWS.** It's your responsibility to safely store the secret access key. It is not stored by AWS.

The function of the secret access key is to prove that you are who you say you are. **AWS keep the access key ID.** You keep the secret access key and you essentially use this secret access key to prove your identity. You cannot ever get back with secret access key from AWS.

what operations you can do with an access key well currently have created it and it's **active**. You can disable an access key and the AWS terminology for that is to make the key **inactive**. It can't be used to interact with AWS. You're able to make an inactive key active again so I can click on Make Active. It will keep the same secret access key, and I'll be able to use it to interact with AWS again. You can also **delete** an access key when you delete it. It's removed entirely from the system, and you can't use it anymore to interact with AWS.

You might face exam questions where you're asked to change the secret access key or you've lost the secret access key, and you need to update the secret access key part for a given set of actors keys. And you can't do that. If you do need to research access keys forever, reason then you essentially need to create a new one. Provide that to any users of that key. So any applications or any command lines and then remove the old one, **you can only have two sets of access keys in a given IAM user**. **Whether the status is active or inactive. You are allowed a maximum of two access keys per AWS user.** Now AWS do recommend that these are **rotated regularly.** So if you have any uses for an access key than what you should do, really is to keep the uses tied down to one of them, and then periodically you can create another access key. You can change all of your command lines or the A P. I related things over to this new access key, and there's a first step. Once you've done that, you can make the old one inactive. You'll be able to see **when the last time this access key was used**. Once it does get used once this will become populated on. Once you see that it's no longer being used, then you can remove it on a DBS. Recommend that you use this process to **rotate these access keys on a regular basis.** **No access keys are checked every time they used.** Whenever an AP I call is made from the command line or using the SDK, you're using a pair of access keys, and every single time that that occurs, you're authenticated. Using those keys and then based on which IAM user those keys belong to you get a set of policies and then you're authorized based on their set of policies. So if you make access keys in active than immediately, every usage of that access key pair is denied. If an access key pair is active, so you're able to use it and you update the policies that are associate ID with the identities of the user, that Access Key belongs to then immediately the permissions that there was access keys have is adjusted.

**Every single time that these access keys pair are used their checked for validity** and then you're authorized based on the policies that the user has at the time that they're used. **Access keys don't expire when you create them.** They work until you make them inactive or you delete them. **There are a type of access keys that do have an expiry**, and we'll talk about them in an upcoming lesson when we discuss I am roles. But it's important to understand that both the username and password and the access keys that are associated with an IAM user are known as **long term credentials**. They don't expire. And for the exam, it's important to remember that you cannot use **access keys to authenticate to the AWS console, so access keys don't let you authenticate to the console** UI and user name and passwords don't let you authenticate using the command line. **For the command line, it's access keys**. **For the console UI It's username and password and optionally MFA**.