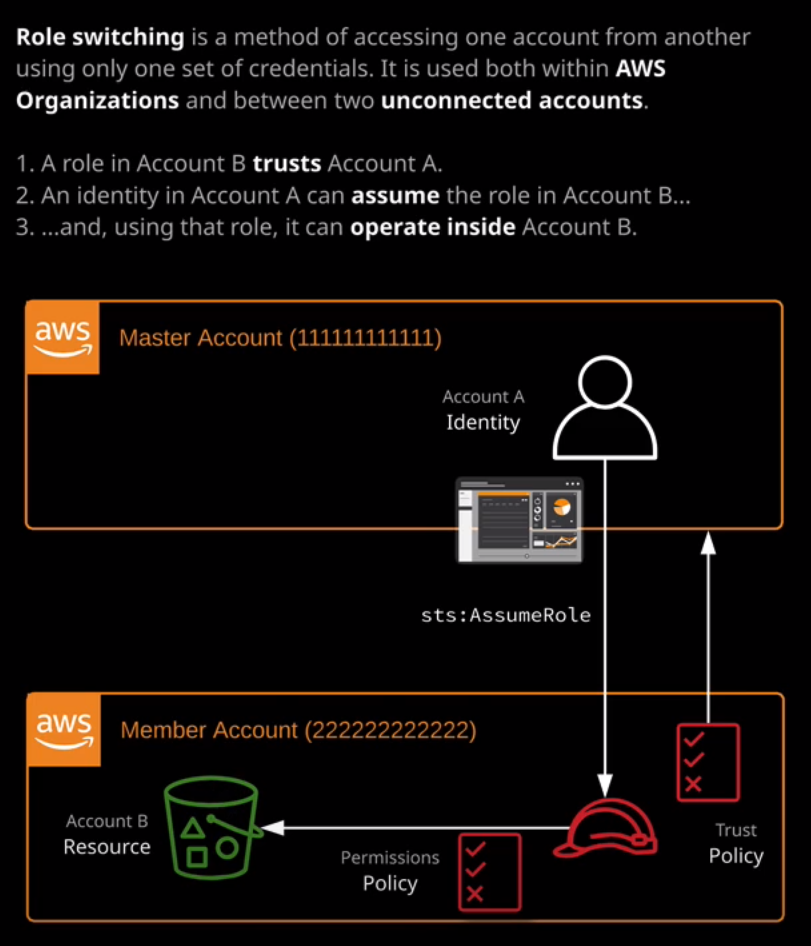
Welcome back and in this lesson I want to talk about a very specific feature of the console UI which is called **role switching**. So role switching can be accessed from the account menu, and it's there, switch role. In the last lesson, we talked about AWS organizations. I talked about consolidated billing, I discuss service control policies and I also briefly touched upon the fact that an IAM role gets created within any AWS member accounts that get created inside the organization. So if you recall in the last lesson, I talked about how, when you click on add account inside an organization, you can either invite an account—an existing AWS account—or you can create a brand new account within the organization. Now, when you do that, there's an IAM role that gets created within that account, and it's called **organization account access role. This role allows us to role switch into that account from this master account,** and in this lesson, I want to talk about how that works. **Role switching is a method of accessing one account from another, using only one set of credentials. Now this is used both within AWS Organizations and it can be configured between two or more unconnected AWS accounts**. The idea is that you've got a single AWS account. In this case, it's the master account within my organization. Inside this account, you've got one or more identities. In this example, it's my IAM user that I'm currently logged in with, which is acantril, and this is the master account of my organization. The idea with role switching is that this IAM user. So this IAM user in my master account can assume a role inside a member account, and using that assumed role can access the console. You I in that member account by only using this centralized IAM user. So it's a way in which you can use roles to switch between the consul. You eye of different AWS accounts. **Now the benefits of doing this is that as an aid to be s organization, you might have three accounts. You might have 10 accounts. You might even have 100 or more accounts. And rather than having to maintain identities in each of these accounts, you can give your user base a single store of identities and allow them to access the other accounts using this role.**



Switch architecture? No. It's far easier just to demonstrate how that works rather than talk about the theory. So that's what I'm gonna do. No. Just to demonstrate that I am actually in the master account, I'm gonna open up the I am console. And inside this master account, I've got my single I am user that I'm using, which is a country. Also note that this a control user is inside the master account. Now, to roll, switch into another account, you need a number of things. First, you need the IAM role created in the account that you are going to role switch into, and that's done by default by a DBS organizations. Well, I also need if I go back to the main AWS console and then go to AWS organizations, I'll need the account number that I want to switch into. In this case for this demo, I'm gonna roll switch into this developer account, so I'm gonna need this account. I d So copy that to my clipboard and to roll switch. You'll need to be logged in as an IAM user and you access the functionality by clicking on the account, drop down and selecting switch role. You'll be given a brief explanation of how it works, but we'll go ahead and click on switch role to begin the process legally to put in your account i d. Of the account that you want to switch into. So this is the account i d of my developer account. You'll also need to provide the name of the role that you're gonna switch into him because we using A B S organizations. This is a standard name. It's called organization Account Access. Roll all one word with uppercase oh, AIDS and are you can provide a friendly display name for this account, So I'm just gonna call it death for development. And then you lied to select a color for this account and actually used within the U I. Just to make it easier to identify, I tend to stick to a scheme where I pick Red for production yellow for development and then, if I have any accounts that he used for experimentation or really proof of concepts are normally pick green or black. So stick to my scheme muppet yellow, and then select switch role. So what this is done at this point is it's made an STS a Schum roll call to the organization account access roll that's in this member account. So this is my development account. So for click on the account drop down and if you ever do this in your account, it might look slightly different. I've actually got a chrome at on installed that extends this functionality, and I'll be priding full details on this later in the course. But the key information will still be there. Even without the sand on, it will show you who he locked in us. So in my case, it's a control inside the L A master account. But it will also show you who you're currently active. As so I'm active is the organization account access role inside this account, which is my development account. Now, to prove that I'm in a different account, I can click on the AWS logo to return to the console and then straight away I'll open the iron console, and if I go toe I am uses, I'll be able to demonstrate that it won't list any because I don't have any use is configured in this account, so you don't see the A control user. And that's because I'm in this member account the developer account now the architecture that this uses. If I click on rolls, you'll know that this account has this organizational account access roll. This was created automatically by eight of US organizations. If I open this up and then click on the trust relationship, stab. This shows the trust relationship with role. I can click on show policy document to show the trust policy for the role and this controls who was allowed to assume this role. This is the account number for the organization Master account. Remember, in the previous lesson, took a standard AWS account that I've created for this cause on I converted it to an organization master account, and this is the account number. So this allows any identity inside the master account to assume this role. And in doing so, anyone who shoes this role gets the permissions that this role has on by default. That's administrator access. Now this is a simple example, but it's the foundation of how simplified user management works. You can have a single account with all of your identities within it on how those identities allowed to potentially roll. Switch into hundreds or more member accounts. Later in this course, you'll also see how you can allow Loggins to this master account, using other forms of identity at the moment of only shown you how you can look into a U. S accounts using I am credentials. So I am users native AWS identities. But it is possible to look in with other forms of identity, maybe a corporate law gin so active directory or other forms of identity Brighter. Later in the course, I'll be demonstrating how you can do that, how you can log into this account using external identities and then hop into other accounts using this role. Switch functionality. Now to do the reverse. To get back to the master account that I deal. Just click on this account, drop down and go back to a control, and that switches me back into the master account. So it takes back the permissions that I've got in the master account, and now I'm logged back in using an identity. In this master account, you'll know how this rolls vanished because it's not present in this master account. And then for Quicken users, you'll see the original identity that used inside the master account to assume that role. So it's a pretty simple technique, but it's one that we can use tow hop between any member accounts inside an AWS organization. Now, the one caution about this is if you are adding additional accounts to an organization so not creating the accounts within the organization for inviting accounts, it's worth remembering that they won't have this. I am role created in that account. If you invite accounts, you'll need to get the account route user or an account. I am user inside that account to create that I am roll. So probably a best practice is to make sure it's called the same so organization account access roll. Make sure that it trusts the master account using the trust policy and then, from that point onward, you can treat it exactly the same as you would as if you created the account inside the organization. So that's essentially it. That's the role switch functionality, and you'll see me use that extensively throughout the course, especially if I'm demonstrating any functionality which requires two different a DBS account So maybe production and development now in the last listener promised to demonstrate the functionality of service control policies, which were a feature of a DBS organizations which can restrict AWS accounts. So I'm gonna do that now. Now, before I do that, I want to demonstrate one thing. So I'm gonna go across and switch into my development account, and I'm just gonna demonstrate that there are no restrictions on this account. So to do that, I'm just gonna move to the sea to console. Remember, this is inside my developer member account, and I'm gonna launch an instance. I'm literally gonna accept all of the defaults because it's not relevant. What the instance Doors. I just want to demonstrate that I can create an instance. So I proceed without a key pair. I'm not gonna connect to this instance It all I'll just demonstrate that can launch an instance. So it's launched in this account with no problems. Okay, so I'm gonna switch back on. I'm gonna move back into my master account. Once another master account. I'm gonna go to a DBS organizations and I'll be creating a new service control policy, so I go to policies And just as an example, I'll highlight this full access policy on go to policy editor. Now, this full access policy grants member accounts The rights to use all eight of us service is if you recall, in effect, this means there are no what restrictions. I'm gonna use this as a foundation for a new policy. Some goto all policies and create a policy. Now, this policy, I'm gonna call s three only so called s3 only I'll use the same name in the description. Now, I can use a policy editor too easily update this policy with the exact service is that I want. But in this case, I'm just gonna paste in this original full access policy on I'm gonna change the actions instead of star. So remember, Starr is a wild card. I'm gonna change it toe s3 colon star. So this is only gonna allow s three actions instead of all actions against all resource is this will only allow s3 action. So any s3 related ap I calls also against any resource is in effect. This will limit only two s three. So I create this policy no one's have created that I need to apply it to our developer account to do that to Goto organize accounts and then that the root of the organization I'm going to select to enable service control policies it'll enable dysfunctionality on enabling. This enables the functionality for the entire organization. What's have enabled service control policies for the entire organization? I could go to the development OU because, remember, it's inside this development OU that the account I'm gonna test with resides in. So I'm gonna select this account, then go to service control policies. There is only one service control policy currently attached to it, Which is this full aid of the US access. Remember, this is the policy that, in effect, doesn't limit the account. I'm gonna attach this new policy. So s three. Only once this one's attached, I'll go ahead and detach the full eight of us access policy. So now for this development account, I've actually got on active service control policy That only allows me to interact using s three ap I calls. So I'm going to demonstrate how this works. I'm gonna go to AWS some things like that to go back to the main part of the 80 best console and then I'm gonna switch in to the development account. So remember and roll switching to an aide to the U. S. I am role that has full admin permissions on this account. So I should have in theory, no restrictions. So let's see how that works. I wanna go too easy, too. And now you can see lots of errors. I'm getting lots of authorization arrows that my user or wherever identity and logged in ours isn't authorized to perform any of these operations. So I'm not authorized to describe running instances. I'm not authorized to describe dedicated host. If a goto launch instances and I try to follow the same process through, I'm getting lots of permissions errors. I'm getting these because now I have this service control policy in place. And so whatever identity I'm logged into this member account with, I won't be able to interact with any service. Is that breach the boundary of this service control policy? If I click on s3, I won't have any problems interacting with this three because the service control policy specifically allows me to interact with s three or specifically It allows me to run any AP I operations, which start with us three. Because this is a necessary I can click on Create Book it for now. We'll just call it a C test 1336 just to make sure it's random except the rest of the defaults on. I'll be able to create this book it without any problems. And that's because the service control policy allows me to do that. Allows me to interact with this account. Using any s3, Cole and Starr actions I'm allowed to interact using s three operations. No, for reverse this process. If a switch back to my master account this time I'm going to go to services, AWS organizations, organized accounts, development OU, select this account, go to service control policies. This time, I'll attach the full AWS access policy and detach the S3 only policy. When I do that, I can switch back to the development account and then if I go back to a service that I was having difficulties with this time EC2, I've got no problems. Now I can see the service and interact with it and even terminate this instance that I created moments ago. So I can go, Yes, terminate. And because I'm using an identity, an IAM role with full admin permissions, and I don't have a service control policy in place, restricting that I'm able to exercise these full admin permissions. So that's just been a really quick demo about how service control policies work at the associate level. You don't need to know about it in much more depth. There are some much more complicated aspects of service control policies which haven't at this point made into the exam. I do keep a constant eye on any new topics which feature in the exam. And so **if service control policies feature in any more technical depth, I will add additional lessons. But for now, all really you need to be aware is that a service control policy essentially provides a boundary a boundary that anyone inside the member account to which it's applied can't breach. So in this member account, even if I logged in as the account root user, the service control policy would still restrict what I'm able to do inside this account. Remember, service control policies don't affect the master account in any way but for any other accounts or any other member account, they apply to every single identity in that account, including the account root user. So if you get any exam questions about how to restrict the ability of a root user in an account to do anything, the only way you've got to do that is service control policies. And you can only use those if you're inside an organization and only then if you've enabled the use of service control policies by enabling it at the root and then applying it either to the root, the OU, or the account itself. Remember, though for the exam it's especially important. The master account cannot be restricted in any way. That's why it's special**. At this point, we've completed this topic. So we finished up with the multi account and organization topic, and that is the last topic inside Section 2 of the course Identity and access control. In the next section of the course, we will be looking at Compute Services within AWS. Now that service is such as EC2, which is a virtual machine service, and Lambda, which is a function as a service product within AWS, we will be looking at the full range of compute services inside AWS. So go ahead, Mark this video as complete. And when you're ready, join me in section three of the course where we'll look at AWS compute.