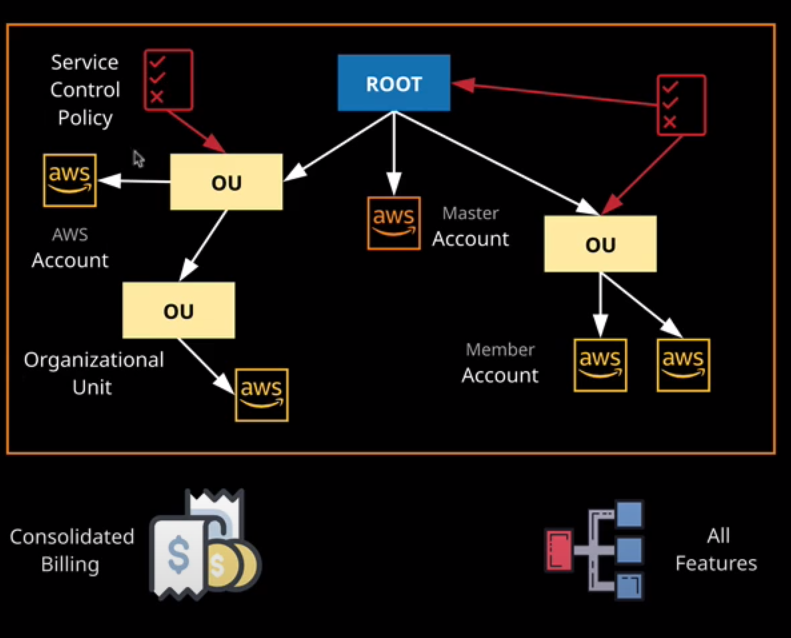


**AWS Organizations are designed to help when managing a large number of AWS accounts. For any businesses with more than 10 accounts, AWS Organizations can be massively useful because rather than having to manage individual bills and individual sets of IAM users, groups, and roles, they could all be managed together.** AWS organizations extends and brings together a number of different legacy pieces of functionality. There's **consolidated building**, which was the old way of allowing multiple AWS accounts to consolidate their billing into one monthly bill. It **adds service control policies which allow AWS accounts to be restricted in multiple ways and it also allows AWS accounts to be created directly from the organization's product.**

AWS Organizations function in two different ways. **The first is just utilizing consolidated billing and what that allows is individual AWS accounts to be combined to generate a single bill.** AWS Organizations has a number of concepts that you need to be aware of. **We start with a single isolated AWS account, and I'm going to turn that a standard AWS account. Every AWS account when you first create it is a standard account**. Now, when you make an organization, what you're essentially doing is **converting that standard account into a master account**. So the first account inside AWS Organizations becomes the master account within that organization. Now, the master account is special. **There can only ever be one master account, and the master account can't be restricted in any way.** So when I talk about service control policies which are ways that you can restrict what accounts can and can't do, the master account cannot be restricted in any way and so it really is important to consider your account architecture. Best practice dictates that the master account should be dedicated to being a master account.

So let's look at how that works. So I'm going to go to services I'm going to go to organizations. Now **it's a global service**, so you don't have to pick a particular region but what you need to do is create an organization. This is the process where the organization gets created. Additionally, this account gets converted from a standard account to being the master account for this organization. So I'll go ahead and do that. I'll create an organization. **Now an organization can operate in one of two modes. The standard mode includes all of the features that's also known as all features mode. That's the one that I'd recommend that you'd use because it brings with it not only the consolidated billing, but in addition, a lot of the extra functionalities that help you administer lots of different accounts. You can elect to use only consolidated billing, but I would recommend that you utilize all features.** So I'm going to go ahead and click on Create Organization. Now you'll be asked to verify the master account. You'll get an email sent to the email address associated with this account, and you'll need to open that email and click on the link that's inside it. At this point, the organization is configured, so any other accounts will automatically consolidate their billing into this master account. Now, the way that this works is if you've got individual accounts that are joined to the organization, they might have their own individual bills. In this example, account one has $10, account two $47, account three $15, and account four $11. Now that means that you've got four accounts with four bills. What you'll find as you proceed throughout this course is that AWS accounts benefit the more you use them. There are certain services such as S3 that become cheaper with the more usage of that product that accounts have. Now, normally, that's not a per account basis but if utilize consolidated billing, it means that the master account receives the bill for the usage of all of the member accounts within the organization. Now that's another term that I need to introduce member accounts. Any accounts that are in the AWS Organization except the master account, are member accounts. **Member accounts pass on their billing to the master account, and it consolidates that billing into a single monthly total.** Now, additionally, and you'll see this through out the course certain services inside AWS such as EC2 allow you to purchase resources in advance. It's called reserved purchases. You do this on a per account basis and if you utilize those products, you benefit from the reduced cost of using those products but it is possible not to utilize reservations in an efficient way. **By utilizing AWS organizations and consolidated billing, it means that any reservations purchased in these individual accounts are combined into this master account and so, as an organization, as a business, you benefit from having a bigger pool of resources**. So anything that you purchase is spread over all of the accounts inside the organization. **So organizations from a financial perspective benefit, from this reduced admin overhead because you only have the single bill, but also you get benefits from the increased usage as well as being able to share any reserved purchases.**



Now also, organizations are operating in the all features mode benefit from what are called service control policies. So this is a structure of an AWS organization. We start off with what's known as the root node and the root node is the top of the organization. Organizations operate with an inverse tree architecture, so we start off with the root at the top. Everything flows down from the root. Now inside the root, of which there is only ever one, you can have AWS accounts, or you can have organizational units and they can contain either accounts or they can contain other organizational units. So you can build up a fairly complex tree style architecture. Now, these service control policies, they can limit what individual accounts can do, so you can define a service control policy to only allow an account to access S3. These service control policies can be applied to individual accounts, to organizational units, or to the root node and they affect any AWS account that is at that level or below it. So if you apply a service control policy to the root node, then it applies to any AWS accounts in that root node or any organizational units in that root node and any organization units in the organizational unit and any accounts and so on. So whatever level you apply these policies they impact that level and anything below it. If you apply a service control policy to a specific account, it only effects that account. If you apply it to an OU it effects that OU so any accounts in the OU and any OUs in the OU and any accounts in the OU and so on.

Now there's one exception to this, and that's the master account. I mentioned this earlier in this lesson. Master accounts cannot be restricted, and that's why it's especially important to decide in advance if you want the master account to be dedicated to just being a master account because you can't restrict it. You might want to restrict AWS accounts to only be able to operate in specific AWS regions, or you might want to restrict what products can be accessed from that account. The key architectural point is that if you do use the master account to host any resources, you won't be able to restrict what gets used inside that master account. **So best practice is that generally this master account is only used for the consolidated billing part of the organization. It can also be used for things like centralized logging. So all of the different accounts in the organization can push their logs into this master account, and it can also be used for IAM users for your entire organization.** So you might have all of your logins occurring to this master account and then roles are used to switch into any other member accounts.

Now, in order to add accounts to this organization, you can do it in two ways. So I'm going to go ahead and click on add account. You can either invite existing accounts into the organization or you can create new accounts directly within the organization. Now, I'm going to go ahead and demonstrate creating an account inside this organizations. I'll click on create account. You'll need to specify a full name for the account, a dedicated email address for the account, and then you'll need to specify the role name. It gets created inside the member account, and it's used to switch into this account from other accounts so **you can log in to the master account using an IAM user and then, rather than having to log in to each of the other individual member accounts, you can simply switch into this account using this IAM role** and I'll be demonstrating that in the next lesson. So I'm going to go ahead and create this account. I'll give it the name LA and then hyphen -saac01 which is the AWS code for this course and then I'll call it Dev for development. The email address that I'll be giving this account is a similar format to what I've used before and then I'll be using my personal email domain. So cantrill.IO. Now I'm going to leave the IAM role as default. Remember, from a security perspective, this role will have a trust policy and the trust policy will allow me to switch into this role from this current account so we don't need to worry about it from a security perspective and we can always adjust it after the fact. So I specified the full name for the account. I have specified the email address to use. I'm going to leave the role as default and hit Create. Now once the account has finished creating I could go ahead and click on organize accounts and begin setting up my organizational structure. I don't have to do that. I can stick to just having the single root code but if I do want to create something a little bit more complex, I could go inside the root node and create additional organizational units. Maybe I want to create one called development so I can type development and create that organizational unit and that will create this additional OU. Maybe I want a new one called production. I'll create that and maybe since I'm creating a course, I want to find one called course development. So I'll name that and then hit Create Organizational Unit. Now I can move individual accounts into these organizational units by selecting the account, clicking move, and then picking the organizational unit to put the account in to. So with this one, which is the development account, I'll select the development OU and then move it into that OU. With the master account, I usually leave these in the root node. Now I can add additional accounts of this organization bucket but there are relevant account limits. ***By default an organization inside AWS can only have a relatively low number of accounts in to start with. The base limit is two accounts inside an organization, so what I always do when creating an organization is to go to the support center and log a service limit related ticket***. So open the console, click on Create Case, go to service limit increase, type organizations, change the limit to the number of accounts, and say that you want to set the new limit to five and then you should normally put in a case description to tell AWS why you want the service limit increase. So I'm going to do that. So I'll put in a suitable message. I'll leave the default that I'm happy to be responded to by web and click subnet and then at this point, the ticket is logged. AWS will review of the service log increase make the change, contact me if required, but at some point over the next couple of days, I'll be able to add additional accounts into this organization. Now, when I'm making training courses I have normally got a fairly standard structure. I create a master account, I create a production and a development account and then I've got a separate account where I do my course development. So I'm going to leave this video for now and I'm going to let AWS increase this limit. I've already got a separate AWS account created for production. Once the AWS limit gets increased I'll add this production account to the organization and then I'll create a second one for course development.

Now, the one thing that haven't talked about **yet SCPs or service control policies. These are the things that you can use to restrict what can happen inside an AWS account. I go to policies, the default policies called full AWS access and this is the policy that essentially adds no limitations to the AWS account. So this is a boundary style policy. It limits what the account actually does and in this case, it allows all actions on all resources. The net effect of this is not to apply any restrictions.** Now I'll be demonstrating service control policies in the next lesson where I'm also going to be demonstrating, role shifting into different AWS accounts but in order to do that, I need AWS to have increased the service limits. So for now, I'm going to finish up this video. We've covered all of the important aspects of AWS Organizations. We've talked about consolidated billing. We've talked about the ability to create AWS accounts inside an organization. We talked about this special case of the master account and that it can't be restricted. I've talked about how service control policies can be applied to these different areas of the organization. What we'll be doing in the next lesson is using an account inside the master account to role shift into a different member account inside the organization and demonstrate how these service control policies can be used to restrict these accounts but for now, that is everything I wanted to cover in this lesson. Go ahead and mark it as complete and I'll see you in the next lesson where I'll cover the rest of the functionality of AWS organizations.