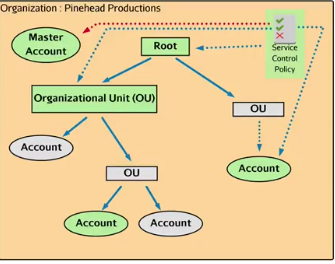
Documentation: How to set up AWS Organizations

I would like to prefix this document and explain what is in it. This describes AWS Organizations and Service Control Policies. This was a big part of the Security Exam the first time I took it and ACloudGuru did not cover it in their course. In this document it also has example questions that tells you how questions around this topic might look like. At the end it also has a documentation section in which I setup my own Organization and test a few things out.

AWS Organizations is a multi-account management system. It allows you to manage multiple accounts within an organization. Organizations presents a hierarchal structure for your accounts. Previously for AWS there was only a consolidated billing feature which allows companies to have one single account for all the billing. Organizations builds upon that. You can set it up for consolidated billing or you can choose all features. All features add service control policies and role switching into other accounts.

Service control policies act as a permissions boundary for accounts in your organization.

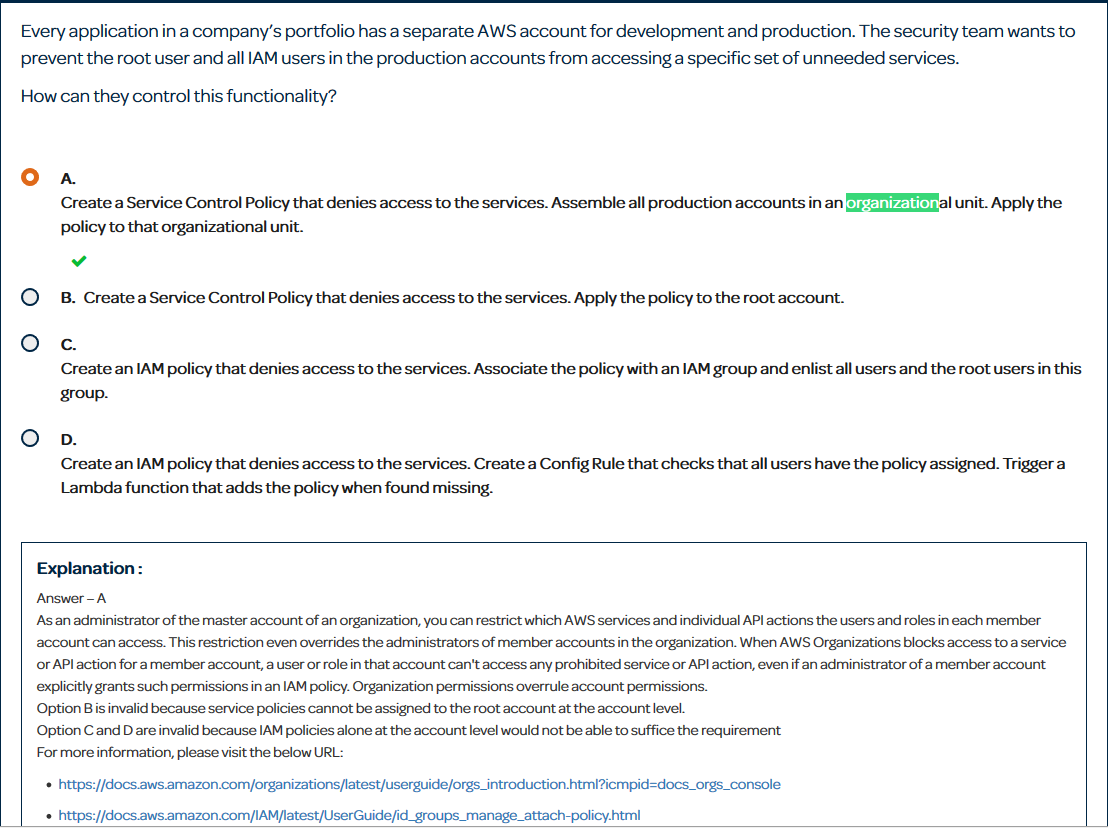


Let me use this diagram to better explain how it works. You start with a master account. That account is the only account with a root user that is unaffected by service control policies. If you add this policy to the master account it trickles down to every other account and user made in those accounts. The root can branch out into organizational units which you can think of as a container for accounts. If you apply the policy to an OU that OU and every other OU/account will be affected by that policy. You can also add service control policies to single accounts. If a policy is added to a single account, not including the root account, that policy only affects that one account.

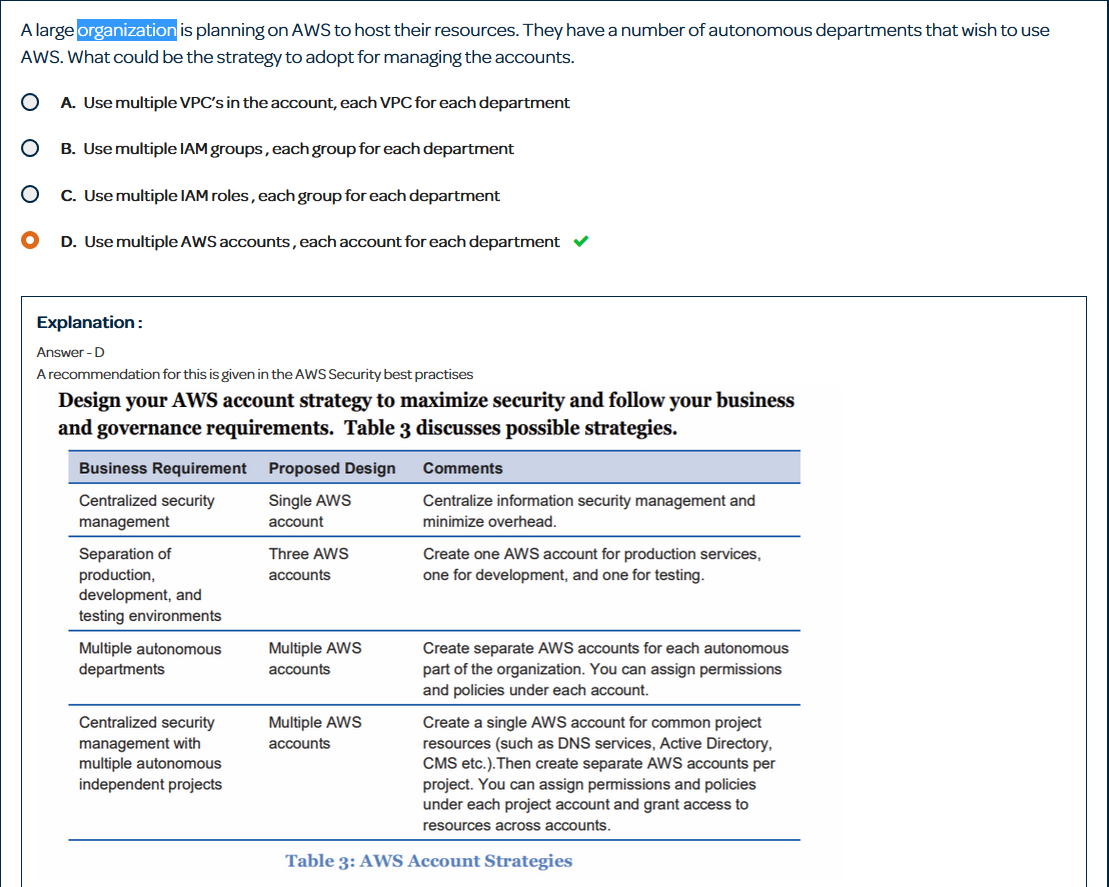
Role shifting is a common way to navigate through your organization. If you choose to add an account by making a new account, you will only be able to access that account via role shifting at first. You would need to role shift into the new account and setup a root password before being able to login.

From the exam the main point you should pay attention too is that you can restrict service access to root users if they are in AWS organizations. You should also understand that in a situation where you have multiple teams, like Dev, Production, Staging, having a multi-account architect is considered best practice from a security point of view. To better explain this pretend one of your accounts got hacked. The hacker might only have access to the dev account thus limiting the blast radius of security incidents. You could have a single account and go VPC by VPC, but that only separates the blast radius from network attacks.

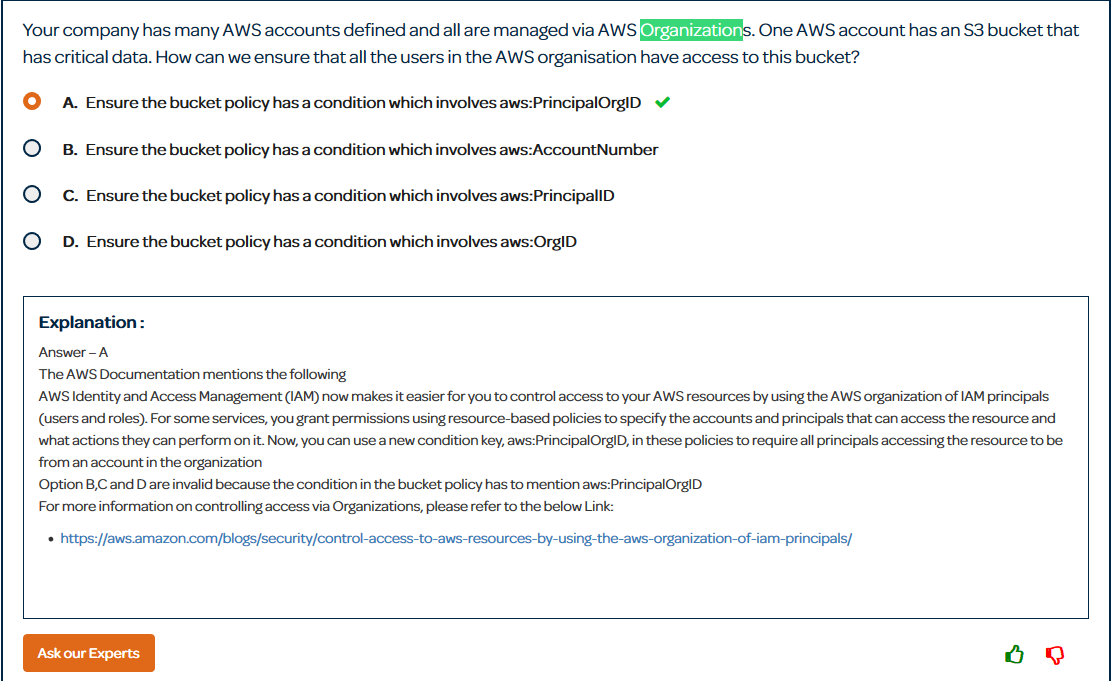
**Example Questions:**



In this question they want to prevent the root and IAM users in the production account from accessing unneeded resources. The reason the answer is not B is because applying a service control policy to the root account in the org will affect all accounts. You only want to deny these services in the production account so the correct answer is A.



Like I explained above, separation by accounts is considered a best practice. You can reference the table for more information.

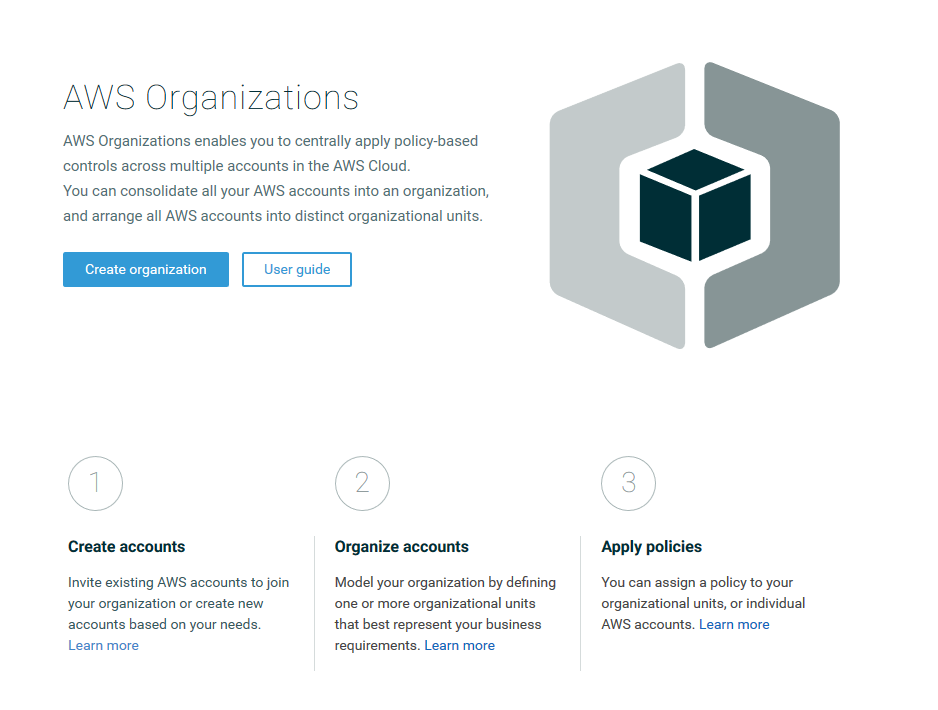


The aws:PrincipleOrgID in a bucket policy allows you to add a condition that only allows access to that bucket from your Organization.

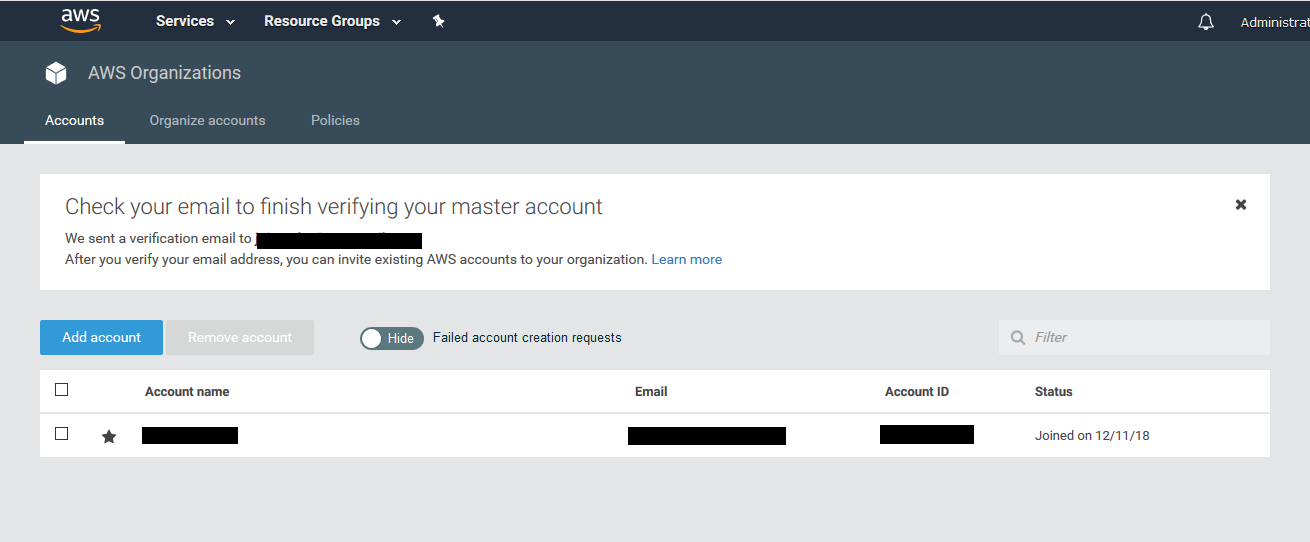
**Setting Up**

**Step 1:**

The first step is to make sure you have an email you can use for a second account. When creating an Organization you can choose to add existing accounts or create a new account granted you have an open email. In your current account locate the AWS Organizations service.



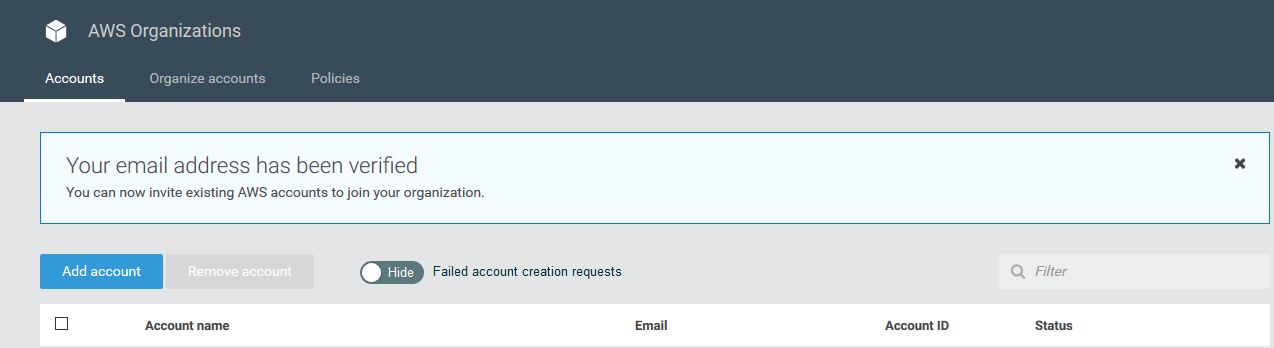
Click create and organization and you will get a pop up telling you about organizations. To continue click create organizations in the pop up.



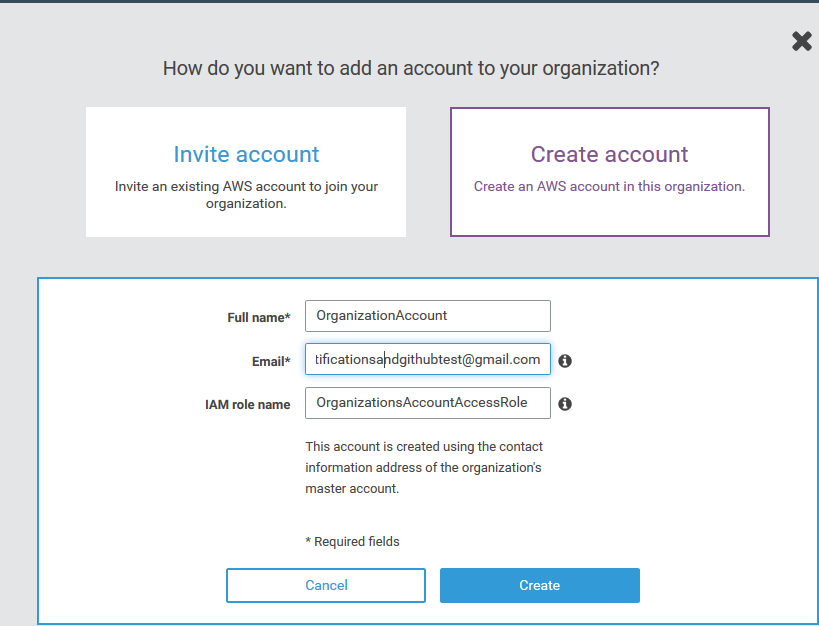
After you create your Organization you will have a screen like this. Notice at the top that you need to verify your email address of the master account. After you verify your email you can move on to the next step.

**Step 2:**

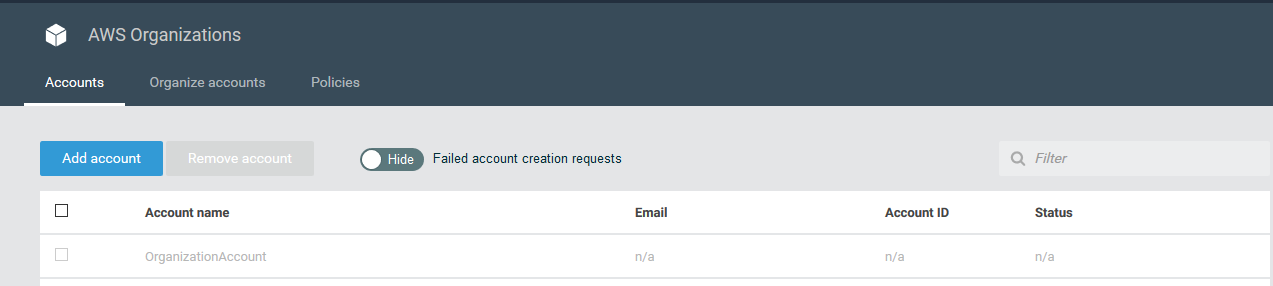
The next step would be to add an account to your Organization.



Click on the blue add account button near the top. You will get 2 options, create account or invite account. I will choose create account as I only have 1 AWS account right now.



When you create a new account you must give it a name, an email not already used, and you must give the IAM role a name. That IAM role will be used to access the new account for the first time. After you filled all of that out click create.



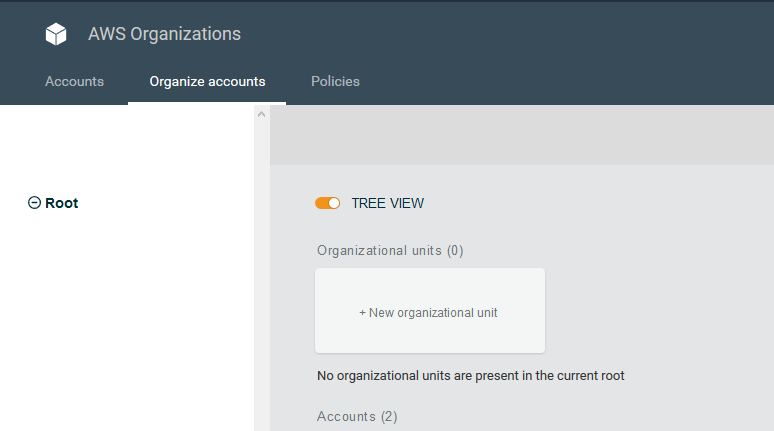
When you add a new account it will start off grey meaning it is not ready for use. After it loads fully you can move onto the next step.

**Step 3:**

Now we want to create an Organizations Unit to make are Organization nice and organized.



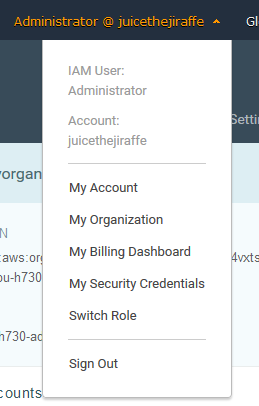
At the top click on Organize accounts.



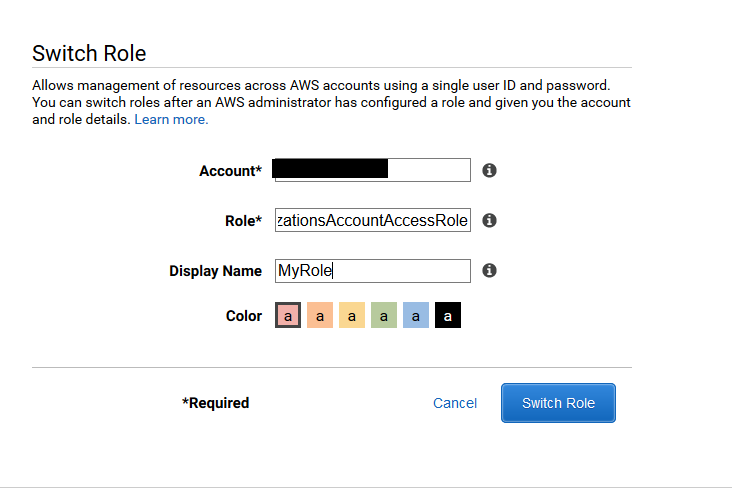
You can then click on new organization unit. It will prompt you to name the organization unit and after you name it you can click create. Next you want to move your new account into the new Organizational unit you just created. You did this by clicking on the account and selecting move. You then select the OU you want to move it to.

**Step 4:**

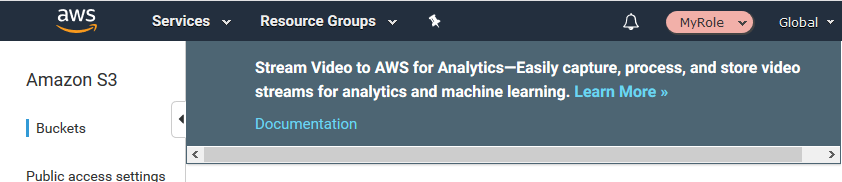
Now we want to access the new account. To do this we click the drop down menu at the top and we click the switch role option.



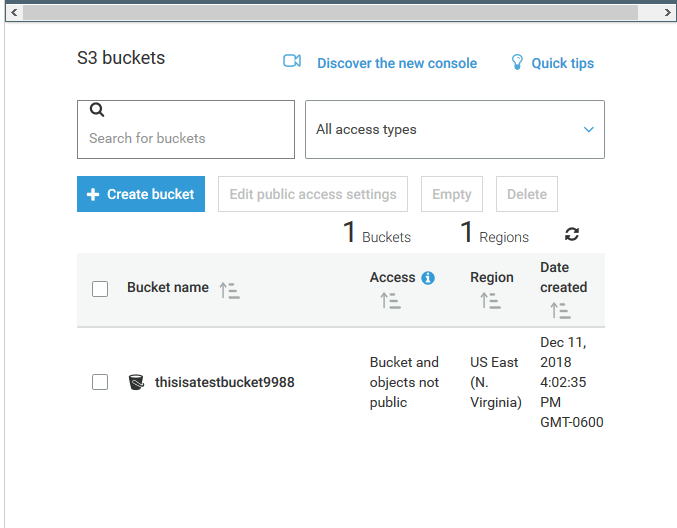
When we decide to switch roles we need a couple of things. We need the account number of the account we are switching into, and we need the role name that we are going to use.



Once you input those you click switch role. You know you’ve succeeded when at the tip it shows the role name you are using. Notice the pinkish MyRole title.

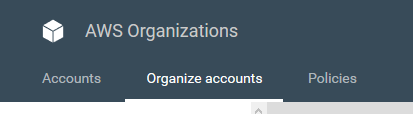


This role by default will have full admin permissions to allow you to set a Root password or configure users. To display service control policies I created a bucket in my second account.

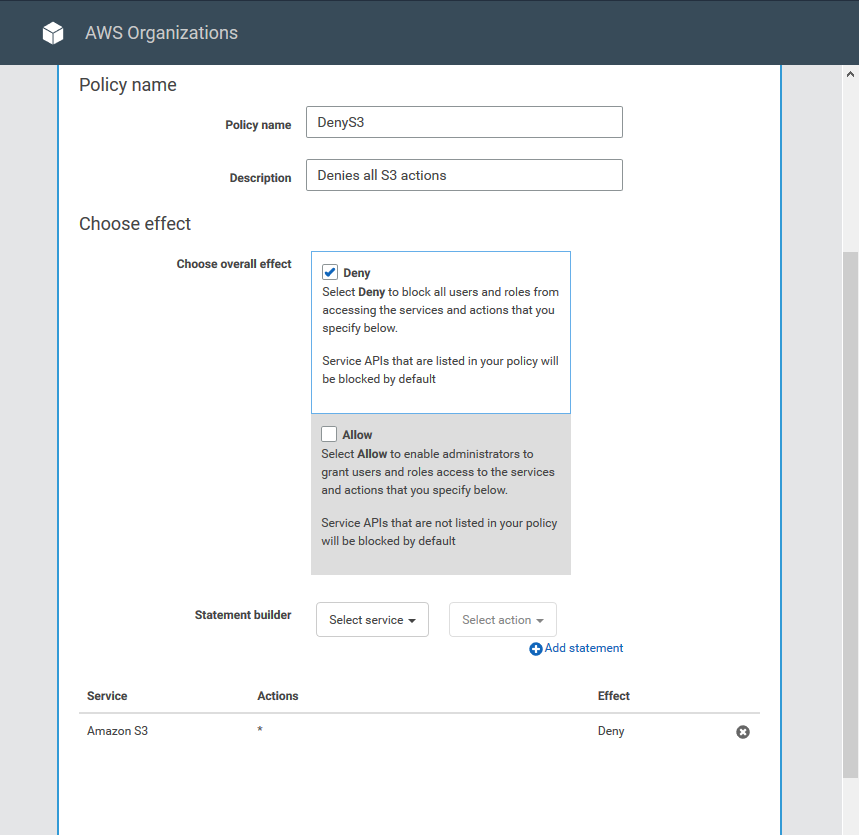


**Step 5:**

Now we need to configure the service control policy. I am going to configure it so that there will be a full access service control policy on the root, but I will have a policy that denies access to S3 on the Organizational Unit I created earlier. This policy would trickle down to every account and every OU below that OU. To do this go back to AWS organizations and click the policies tab at the top.



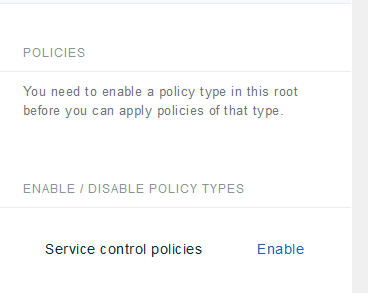
Click create policy and you will get an easy to navigate GUI for making IAM policies. For this policy I am just going to deny all S3 actions.



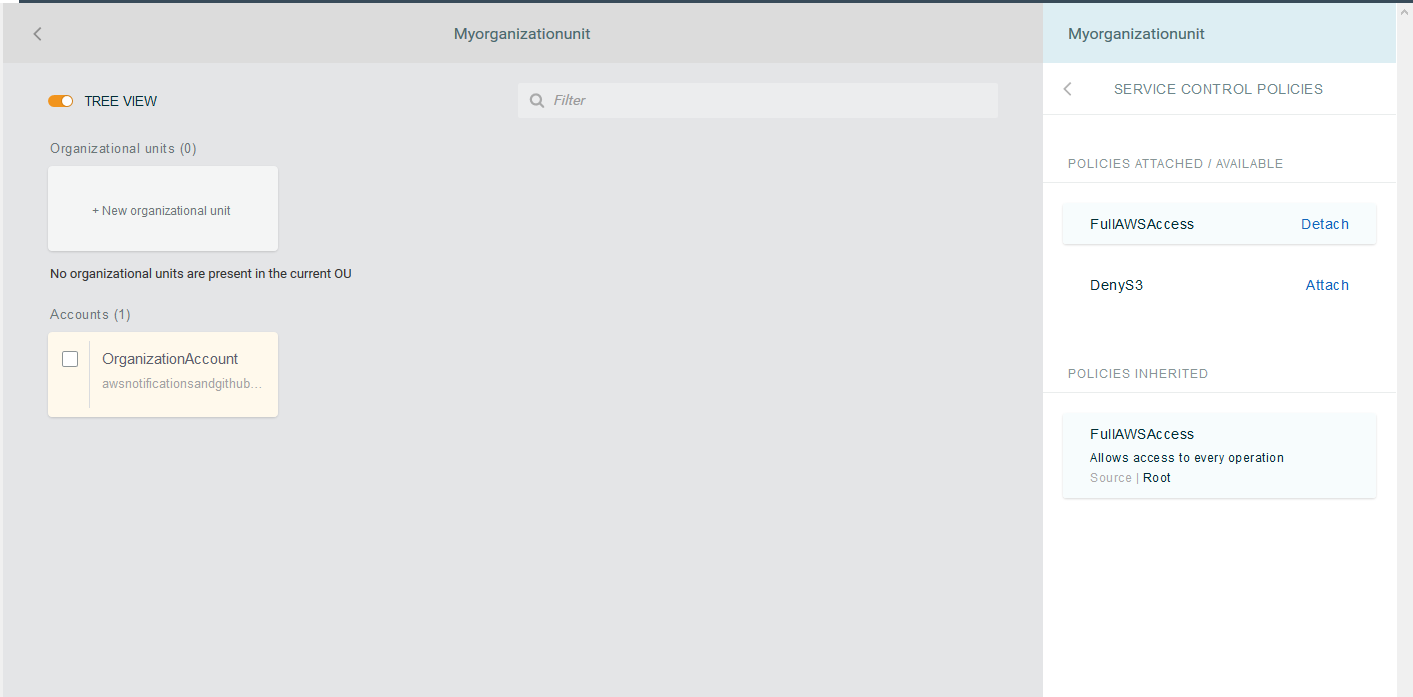
Click create policy and go to organize accounts at the top.

**Step 6:**

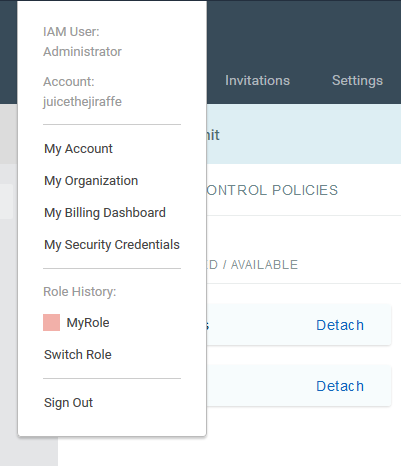
Before you can attach a policy to an OU you must first attach a policy to the root account. Luckily they create a full access policy when you make your org. On the right hand side locate the enable/disable policy types and click enable.



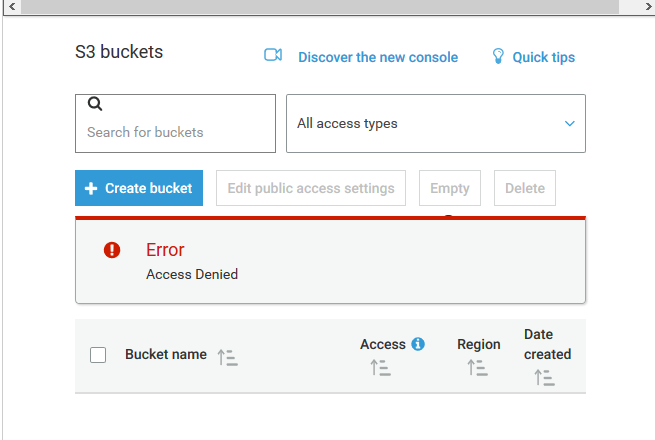
After you enable it click on the organizational unit you created earlier.



Remember because every policy on the root trickles down, you should have the full AWS Access policy already attached. Click attach by the DenyS3 policy and role shift into the other account.



When you get into the other account navigate to S3 to test out the policy.



Keep in mind that this policy will also deny access to S3 on the root user of this account. This is the only way to restrict root user access. That may come up in the exam.