AWS ORGANIZATION LAB

Prerequisite:

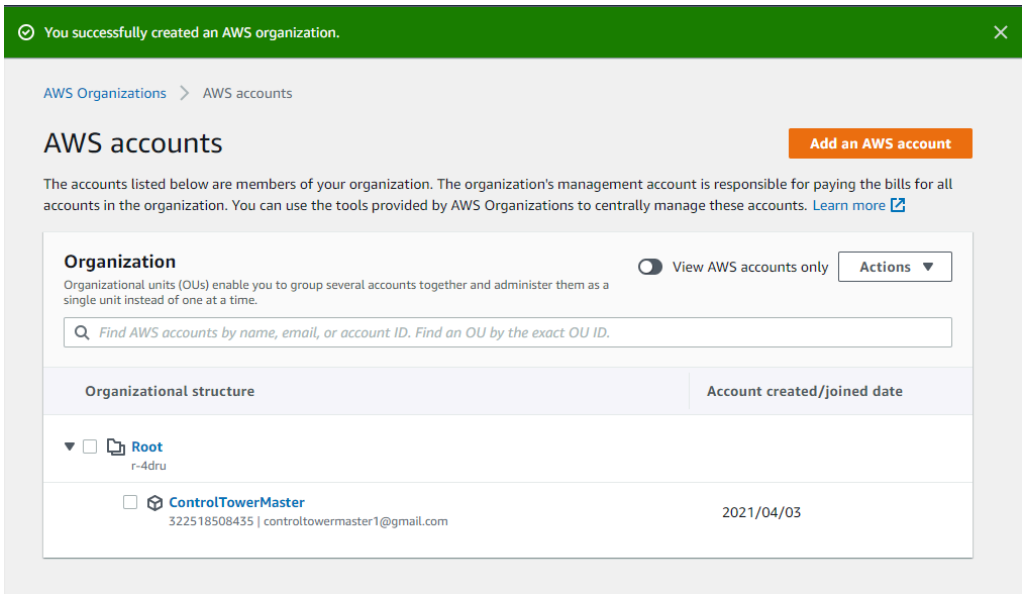
1. Have an AWS account which will be the management account
2. Have an email address not tied to any AWS account. This email will be used to create an AWS account from within the organization

Tasks:

1. Create an AWS organization with the current account as the management account
2. Create another account from within your organization that will be automatically added to your organization.
3. Use switch role to move from one account to another
4. Use SCPs to establish guardrails/restrictions of what is allowed and denied
5. Test restrictions

LAB1: CREATE AN AWS ORGANIZATION

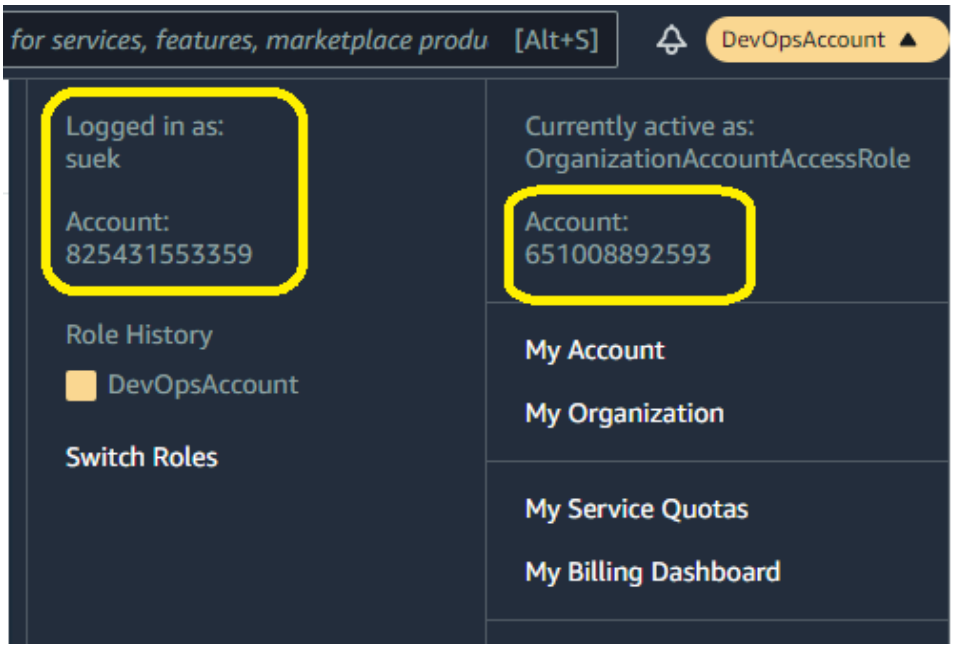
1. Log into the AWS account which will be considered the master account as an IAM user with Admin privileges.
2. Click on 'create an organization' to create an AWS organization in the master account. Once the organization is created you will see the root container with the master account in it



3. After the Organization is created, you should receive an email in the management account to verify your email address before you can invite existing AWS accounts to join your organization.

4. Click on Add an AWS account. *You can either invite an existing account or Create a new account which will automatically be added to the organization.*
    a. Enter the parameters to create a new account.
        i. For Account Name use **JJTECH-DevAccount**
        ii. You can leave the IAM role name blank to automatically use the default role name from AWS (***OrganizationAccountAccessRole***) or you can add your own role name
    b. Choose Create AWS account This account will be created for you and automatically added to your organization

5. You can access the newly created account by either using the IAM switch role that was created above or using the root user credentials that was created by the organization.
    a. To access with switch role:
        i. Open the AWS Management Console using IAM user credentials.
        ii. Choose your account name at the top of the page, and then select Switch Role. Important: If you are signed in with root user credentials, you can't switch roles. You must be signed in as an IAM user or role. For more information, see Switching to a role (Console).
        iii. Enter the account number and role name for the member account.
        iv. (Optional) You can also enter a custom display name (maximum 64 characters) and a display color for the member account.
        v. Choose Switch Role and you will be taken to the new AWS account while logged into the master account. (See screen below).

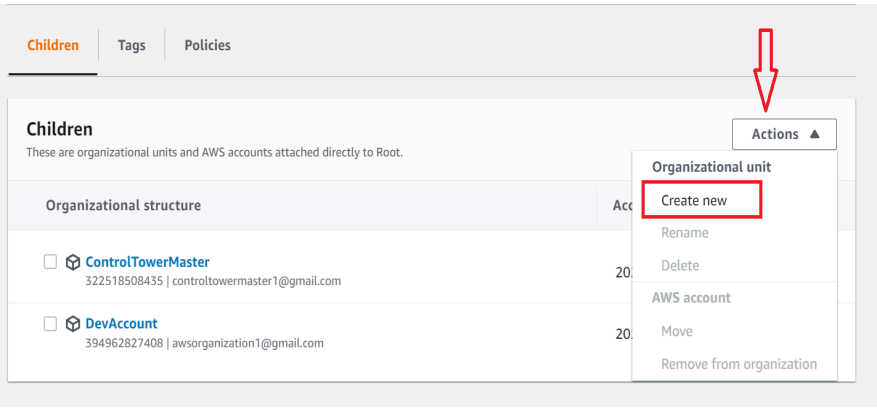


    b. **To retrieve the root user credentials:**

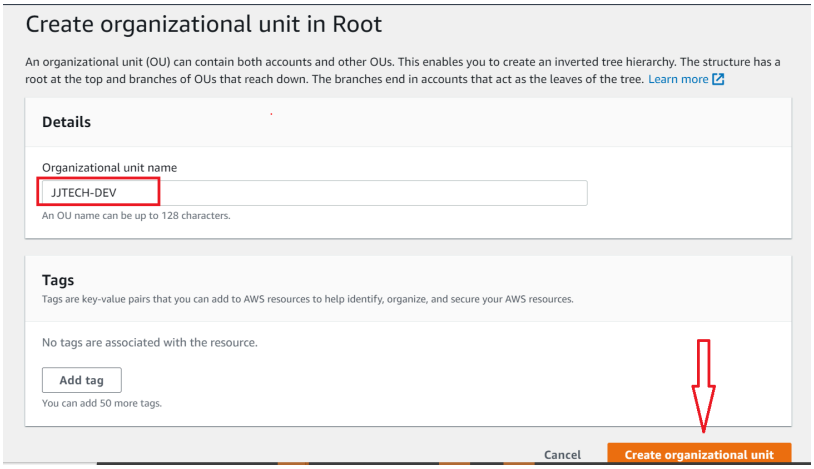
i. Go to the sign in page of the AWS console and choose Sign in using root account credentials
ii. Choose Forgot your password? and enter the information that is required to reset the password to a new one.
iii. Check your email and choose the reset password link.

LAB2: CREATE ORGANIZATIONAL UNITS

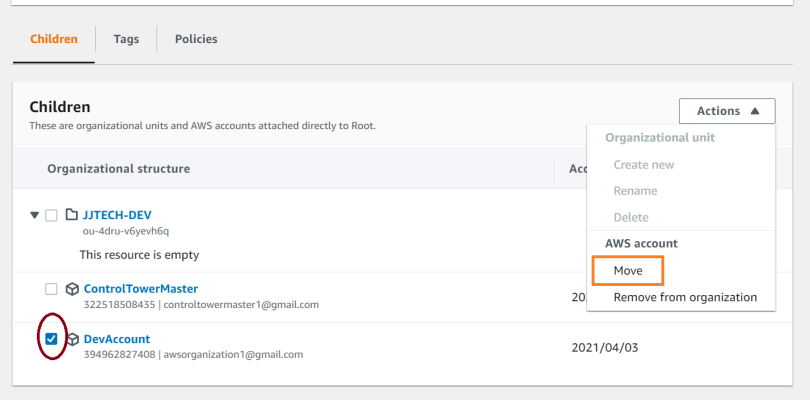
1. Go to the AWS organizations console and navigate to the AWS account page
2. click on the root container, choose actions and then under Organizational unit, choose create new.



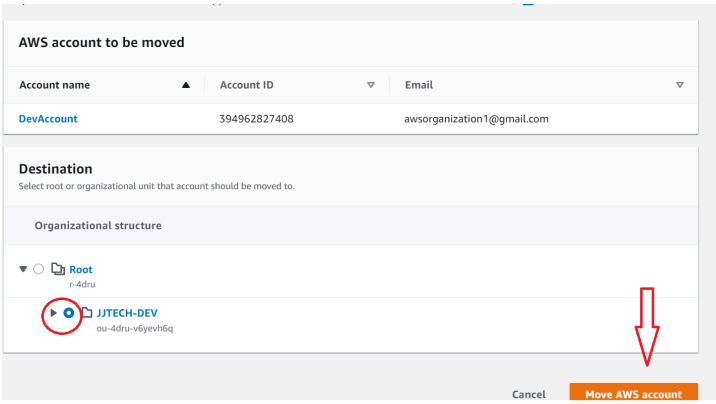
3. Enter the organizational unit name *JJTECH-DEV* and select Create organizational unit.



4. Select the member you just created from within the AWS organization, choose Actions, and then under AWS account, choose Move.



5. On the *Move AWS account 'DevAccount'*page, choose JJTECH-DEV OU and then choose Move AWS account.

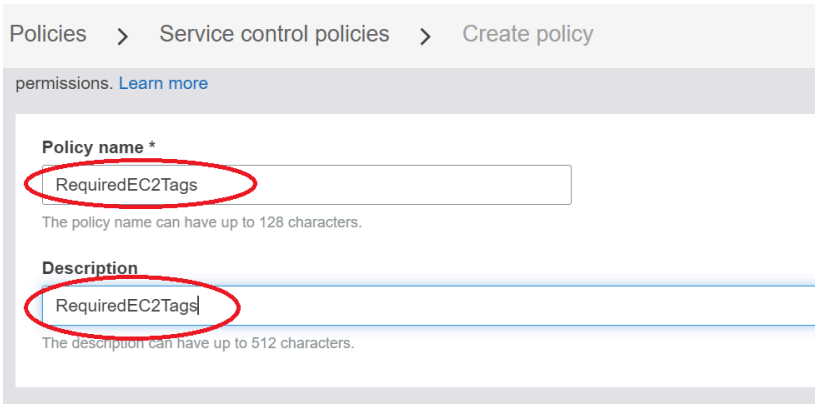


LAB3: ENABLE AND CREATE SERVICE CONTROL POLICIES

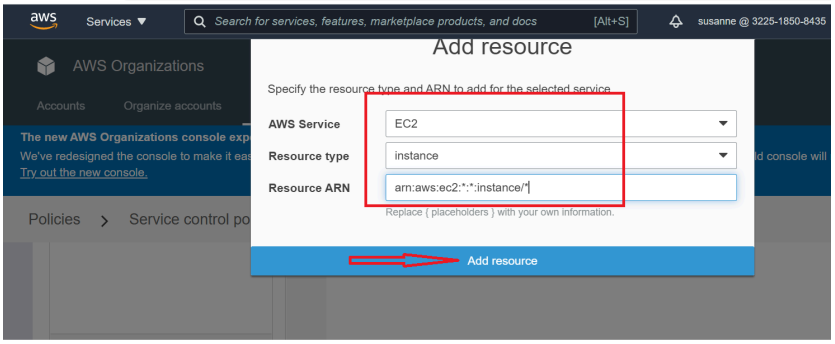
1. Navigate to the policies page, and then choose Service Control Policies.On the Service control policies page, choose Enable service control policies.

   A green banner appears to inform you that you can now create SCPs in your organization.

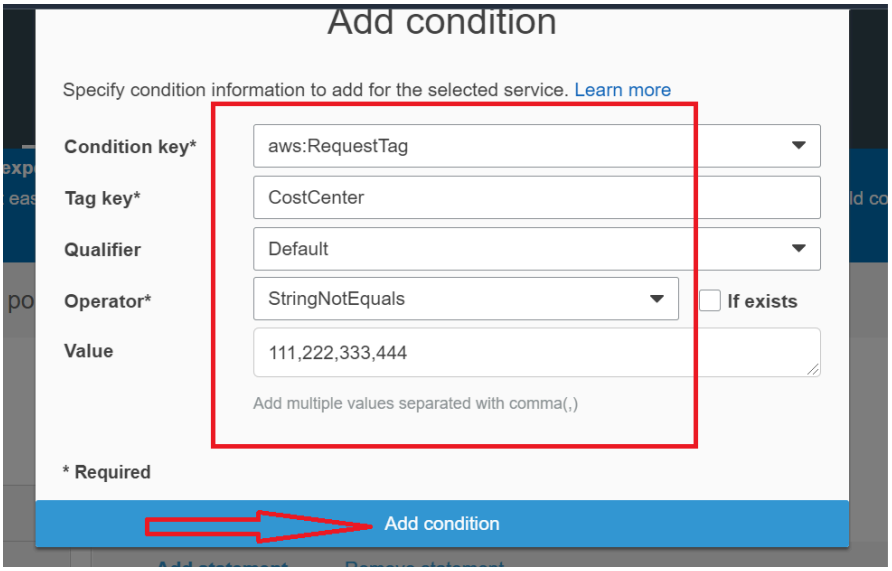
2. Choose create policy to create a policy. Enter RequiredEC2Tags as the policy name and description.



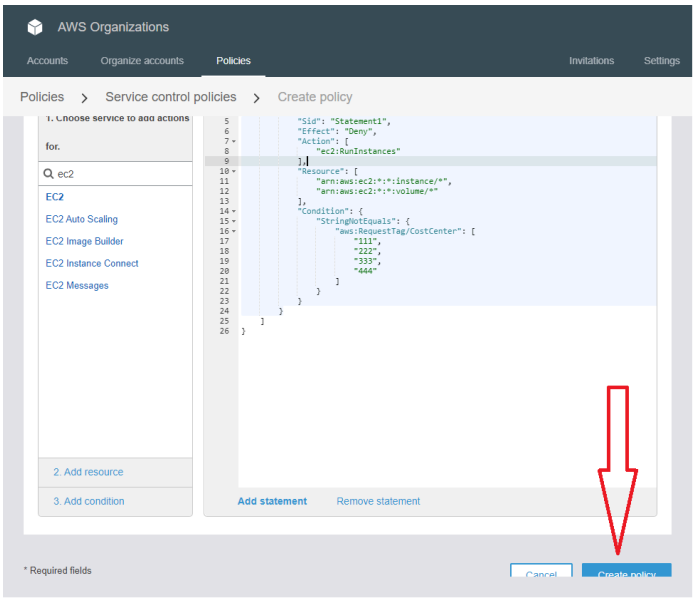
3. Next Enter the parameters for your policy statement details
    a. Choose service to add action:
        i. For services, search **EC2** in the filter
        ii. For action, search **RunInstances** in the filter
    b. Select Resources within the service:
        i. Click on **Add resource** and there the below parameters to create an EC2 instance resource. AWS Service = EC2, Resource Type = Instance, Resource ARN = Replace all curly brackets parameters with an asterisk (*) and it should look like this *arn:aws:ec2:*:*:instance/**
        ii. Click Add resource



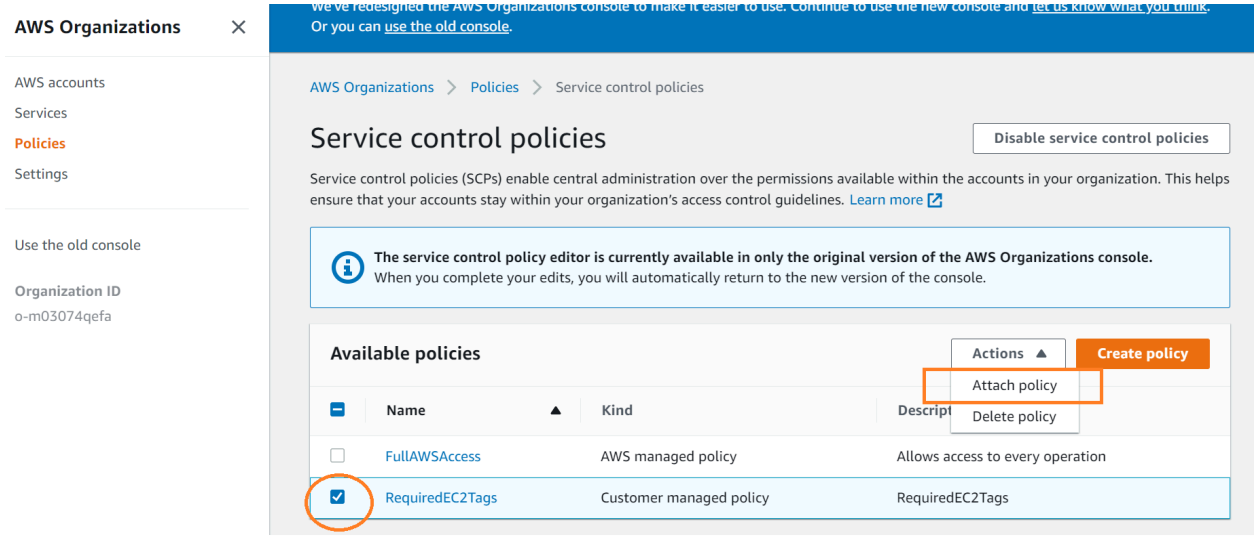
        iii. Repeat step d above for a **volume** resource type. This means we will be enforcing this policy on both instance and volumes. *(arn:aws:ec2:*:*:volume/*)*
    c. Add condition:
        i. Click on **Add condition** and select the conditionkey aws:RequestTag
        ii. Tag key = CostCenter, Qualifier = Default, Operator = StringNotEquals, Value = 111,222,333,444
        iii. Click on Add condition



4. Click on create policy



5. Select policy - Go to Actions and click on Attach policy



6. Attach policy to the JJTECH-Dev OU
7. Test policy by logging in as a root user into a member account in the JJTECH-Dev OU, Go to any region and test the following
    a. Create an EC2 instance with no tag (this will fail)
    b. Create an EC2 instance with a tag key and no value (this will fail)
    c. Create an EC2 instance with tag key cost-center and value 11(this will fail)
    d. Create an EC2 instance with tag key CostCenter and value 111 (this should work)

Create other SCPs with other policies below and test the effectiveness

1. Prevent users from deleting VPC flow logs and Cloudwatch log groups

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteFlowLogs",
        "logs:DeleteLogGroup",
        "logs:DeleteLogStream"
      ],
      "Resource": "*"
```

```
    }
  ]
 }
```

2. Prevent any VPC that doest already have internet access from getting it

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:AttachInternetGateway",
        "ec2:CreateInternetGateway",
        "ec2:CreateEgressOnlyInternetGateway",
        "ec2:CreateVpcPeeringConnection",
        "ec2:AcceptVpcPeeringConnection",
        "globalaccelerator:Create*",
        "globalaccelerator:Update*"
      ],
      "Resource": "*"
    }
  ]
}
```