Table of Contents

[High level objective: IAM User, Group & Policy 1](#_Toc37298817)

[High level objective: IAM Role and Role Switching 27](#_Toc37298818)

[High level objective: IAM Organization & Service Control Policy 35](#_Toc37298819)

# High level objective: IAM User, Group & Policy

* Create 3 IAM users under an account
  + admin
  + developer\_1
  + tester\_1
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

* Create two groups
  + devlopers
  + testers
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
|  |
| Create testers group in exact same way. Screenshots are omitted intentionally |
|  |

* Assign AWS Managed policy Administrator to admin IAM user
  + Done during admin user creation
* Create Customer Managed Policy EC2FullAccess which will allow all access to EC2
  + SCREENSHOTS

|  |
| --- |
|  |
| Corresponding JSON  {  "Version": "2012-10-17",  "Statement": [  {  "Sid": "EC2FullAccess1",  "Effect": "Allow",  "Action": "ec2:\*",  "Resource": "\*"  }  ]  } |
|  |

* Create Customer Managed Policy S3FullAccess which will allow all access to S3
  + SCREENSHOTS

|  |
| --- |
|  |
| Corresponding JSON  {  "Version": "2012-10-17",  "Statement": [  {  "Sid": "S3FullAccess1",  "Effect": "Allow",  "Action": "s3:\*",  "Resource": "\*"  }  ]  } |
|  |

* Create Customer Managed Policy S3ReadOnlyAccess which will allow only read access to S3
  + SCREENSHOTS

|  |
| --- |
|  |
| Corresponding JSON  {  "Version": "2012-10-17",  "Statement": [  {  "Sid": "S3ReadOnlyAccess1",  "Effect": "Allow",  "Action": [  "s3:GetAccessPoint",  "s3:GetLifecycleConfiguration",  "s3:GetBucketTagging",  "s3:GetInventoryConfiguration",  "s3:GetObjectVersionTagging",  "s3:ListBucketVersions",  "s3:GetBucketLogging",  "s3:ListBucket",  "s3:GetAccelerateConfiguration",  "s3:GetBucketPolicy",  "s3:GetObjectVersionTorrent",  "s3:GetObjectAcl",  "s3:GetEncryptionConfiguration",  "s3:GetBucketObjectLockConfiguration",  "s3:GetBucketRequestPayment",  "s3:GetAccessPointPolicyStatus",  "s3:GetObjectVersionAcl",  "s3:GetObjectTagging",  "s3:GetMetricsConfiguration",  "s3:HeadBucket",  "s3:GetBucketPublicAccessBlock",  "s3:GetBucketPolicyStatus",  "s3:ListBucketMultipartUploads",  "s3:GetObjectRetention",  "s3:GetBucketWebsite",  "s3:ListAccessPoints",  "s3:ListJobs",  "s3:GetBucketVersioning",  "s3:GetBucketAcl",  "s3:GetObjectLegalHold",  "s3:GetBucketNotification",  "s3:GetReplicationConfiguration",  "s3:ListMultipartUploadParts",  "s3:GetObject",  "s3:GetObjectTorrent",  "s3:GetAccountPublicAccessBlock",  "s3:ListAllMyBuckets",  "s3:DescribeJob",  "s3:GetBucketCORS",  "s3:GetAnalyticsConfiguration",  "s3:GetObjectVersionForReplication",  "s3:GetBucketLocation",  "s3:GetAccessPointPolicy",  "s3:GetObjectVersion"  ],  "Resource": "\*"  }  ]  } |
|  |
|  |

* Assign EC2FullAccess & S3ReadOnlyAccess to developers group
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
|  |

* Assign S3FullAccess to testers group
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
|  |

* Add developer\_1 user to developers group
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
|  |

* Add tester\_1 to testers group
  + SCREENSHOTS

|  |
| --- |
| Follow the steps of adding developer\_1 to developers group. Screenshots are omitted intentionally |

* Login with admin, developer\_1 & tester\_1 in Admin Console and test whether they have their expected level of access
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
| ☹ This is a bummer but no big deal |
|  |
|  |
|  |
|  |
| Saw no error as admin and had access to all AWS services. Screenshots are omitted intentionally. Log out as admin |
|  |
| No IAM Access |
| Full EC2 Access |
| Can visit S3, but can not create Bucket. At this point, logout as developer\_1 |
|  |
| No IAM Access |
| Can not create EC2 instances |
| Can create S3 bucket. At this point, logout as tester\_1 |

# High level objective: IAM Role and Role Switching

* Create a role to allow developers temporarily assume admin role
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
| At this point, take a note of following details:   * Account Id: 706960460656 * Role: temporary-admin-role * Sign in URL: https://signin.aws.amazon.com/switchrole?roleName=temporary-admin-role&account=706960460656 |
| Now, we need to create a policy which allows to assume the role that we just created. We will assign this policy to developer group |
|  |
| Corresponding JSON  {  "Version": "2012-10-17",  "Statement": [  {  "Sid": "TempAdminRole1",  "Effect": "Allow",  "Action": "sts:AssumeRole",  "Resource": "arn:aws:iam::706960460656:role/temporary-admin-role"  }  ]  } |
|  |
|  |
|  |
| Logout as admin user |

* Test whether developer\_1 can become an admin by role switching
  + SCREENSHOTS

|  |
| --- |
|  |
| As expected, do not have IAM access. Now switch role |
|  |
|  |
| After switching role, developer\_1 has full IAM access |
| Now, switch back to developer\_1 and logout |
|  |
|  |

# High level objective: IAM Organization & Service Control Policy

* Note: We do not enough permission in linux academy playground. Completed the exercise using personal AWS account
* Create an organization under an AWS account
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
| Can not create organizations in Linux Academy. Used my personal AWS account for this activity |
|  |
|  |
|  |

* Add a new account under that organization
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
| IMPORTANT : Ensure the email id entered has no prior AWS accounts. Otherwise the organization creation will fail |
|  |

* Create a service organization under root organization and assign the new account under that organization
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

* Create a service control policy that restricts access to S3 only
  + SCREENSHOTS

|  |
| --- |
|  |
|  |
|  |
|  |
| Corresponding JSON  {  "Version": "2012-10-17",  "Statement": [  {  "Sid": "Statement1",  "Effect": "Allow",  "Action": [  "s3:\*"  ],  "Resource": [  "\*"  ]  }  ]  } |
|  |

* Enable that service control policy and attach that to the new Service Organization & the new account
  + SCREENSHOTS

|  |
| --- |
|  |
|  |

* Test whether if the admin user from Master account switches role to the newly created account, whether the access is restricted
  + SCREENSOTS

|  |
| --- |
| Get following details:   * Account Id of Secondary Organization: 393478524736 * Role: OrganizationAccountAccessRole (default) |
| Now, will go to IAM 🡪 Create an admin user 🡪 Login with admin user 🡪 Check whether admin user has full IAM access. Now will try to switch role to the Secondary Organization. Before logging in as admin user collect the AWS root user account # |
|  |
|  |
| No access to IAM in secondary Organization. Now will switch back to admin role |
| And has got IAM access back |