

IBM Cloud – Identity and Access Management

A beginner's guide to understanding IAM

By James Belton

Table of Contents

Introduction	3
Basic Concepts.....	4
User	4
Service ID	4
Resource	4
Resource Group.....	4
Access Group.....	4
Access Roles and Access Policy	4
Managing Resource Groups.....	5
Viewing Resource Groups	5
Renaming a Resource Group.....	5
Create a Resource Group	5
What is a good Resource Group Strategy?	6
Adding Resources to a Resource Group.....	7
What permissions (Access Policies) are required?.....	7
Using an Access Group	8
How to Create an Access Group.....	8
Assigning Users and Service IDs to Access Groups.....	8
Adding Service IDs to an Access Group	9
Assign Access Policies to an Access Group	10
How Roles make Policies that define Access	12
What are the roles and what do they do?	12
Platform Management Roles	12
Service Access Roles	14
Combining Access Groups with Identity Federation	15
Two Examples	17
Scenario 1 – Developer Access to an Object Storage Bucket.....	17
Scenario 2: Three Teams working on Three Different Projects.	20
Try Experimenting!	22

Introduction

Identity and Access Management (IAM) is a security model used by an increasing number of IBM Cloud services. It provides users and services fine-grained access to resources within an IBM Cloud account and it is important that account administrators and managers understand both the basic concepts as well as how to implement IAM in their account.

I have written this beginner's guide in an attempt to provide something which is a reasonably user-friendly, describes the basics and works up to a point where it's possible to describe IAM at a basic level, start to be able to piece together a resource group strategy along with an access management strategy.

Remember that IAM is distinctly different to Cloud Foundry access management, which is controlled via access to Organizations and Spaces – however, if you are looking for a comparison, think of a resource group as being logically similar to a Cloud Foundry space.

Don't be fooled into thinking this is an exhaustive guide! While IAM isn't difficult (or at least you shouldn't need a PhD to use such things), things do change from time to time, since that is the nature of cloud. Think of this as a introduction and as you get started and more experienced, refer to the official documentation at [cloud.ibm.com](https://cloud.ibm.com/docs) too.

I hope this gets you started and as time goes by and my knowledge increases, I'll strive to update any bits that prove inaccurate and also add to the examples section too.

Since I wrote the first version of this guide, I have also created a number of videos around IAM and IBM Cloud in general, which I call the IBM Cloud Foundation Skills Series. If you are interested in checking those out, please see <https://www.youtube.com/playlist?list=PLmesOgYt3nKCfsXqx-A5k1bP7t146U4rz>

Regards

James Belton
July 2018
(updated October 2019)

Basic Concepts

These are the basic terms that you need to understand for IAM. If you know and get familiar with these, that's a great start.

User

A user represents an IBMid-enabled account that can access IBM Cloud via a user ID, which is normally in the form of an email address. Users are 'invited' by the account Administrator or those with suitable account level rights. A user can be a real person, or in some cases it can be a task ID. Task IDs can be useful where an account is needed for the lifetime of a project (for admin tasks, for example), since this can persist, while real users may come and go. IAM permissions can be granted to a User ID.

Service ID

A service ID identifies a service or application in a way that is similar to how a user ID identifies a user. A service ID that you create can be used to enable an application outside of IBM Cloud to access to your IBM Cloud services. IAM permissions can be granted to a Service ID.

Resource

A resource is an application, service instance, container cluster, storage volume or virtual server, which can be managed using IAM. When provisioning a resource, it must be placed into a 'Resource Group'. A resource cannot be placed into more than one resource group.

Resource Group

A Resource Group is a logical container for resources. When a resource is provisioned, a Resource Group name must be provided and once provisioned, a resource cannot change its Resource Group. It is possible to have just one Resource Group in an account and place all account resources into it, but it is good practice to have multiple Resource Groups. A recommended model for Resource Groups is to have one per project environment. Note that resource groups have no geographical / region binding and that resources from multiple regions can exist in a single resource group.

Access Group

An Access Group is a collection of User IDs and / or Service IDs to which IAM permissions can be granted. If a number of users or service ID require identical access, rather than grant those IAM permissions multiple times on a user-by-user (or service-by-service) basis, it is quicker and easier to put those users into an Access Group and grant them once to the group.

Access Roles and Access Policy

Access is granted via Access Roles (or 'roles' for short) and an Access Policy (or 'policy' for short) is a collection of roles. Together they determine the level of 'permission' that is granted to provide access to IAM managed resources. There are a number of roles, which give different levels of access, depending on how users or services need to interact with different resources. When creating an access policy from roles, it is recommended that a model of 'least permission' is followed.

Managing Resource Groups

When creating an IAM-enabled resource, IBM Cloud will require the name of a Resource Group at the provisioning stage. Every IBM Cloud account has a pre-created Resource Group, named ‘Default’. While it is acceptable to use this Resource Group and place all resources into it, creating multiple Resource Groups – for instance, one per project environment - will ultimately make resource management easier and is the recommended approach.

A resource group is a logical container, it does not physically reside in any particular IBM Cloud region. This means that resources from multiple regions can be placed in a single resource group.

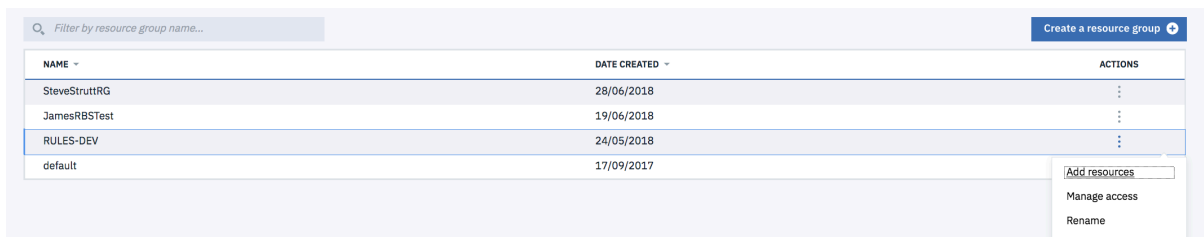
NOTE: You can rename a resource group, but you cannot delete a resource group once it has been created, however, you can rename them.

Viewing Resource Groups

To view the resource groups which have been created in an IBM Cloud account, click Manage -> Account -> Resource Groups. This will display a list of resource group names. At a minimum, there will be one resource group, possibly named ‘Default’, though this may have previously been renamed by an administrator.

Renaming a Resource Group

To rename a resource group, access the list of resource groups by clicking Manage -> Account -> Resource Groups.



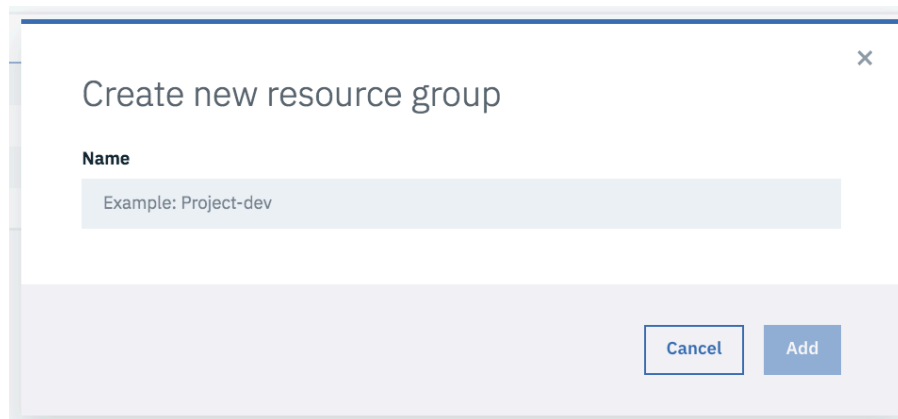
NAME	DATE CREATED	ACTIONS
SteveStruttRG	28/06/2018	⋮
JamesRBSTest	19/06/2018	⋮
RULES-DEV	24/05/2018	⋮
default	17/09/2017	⋮

[Add resources](#)
[Manage access](#)
[Rename](#)

Click the three dots and select Rename from the menu. Type the new name in the pop-up box and click Save.

Create a Resource Group

To create a resource group, in the IBM Cloud Console click Manage -> Account -> Resource Groups. Then click ‘Create a Resource Group’.



Provide a name for the resource group in the pop-up box and click ‘Add’.

What is a good Resource Group Strategy?

Since a Resource Group is a logical container for resources, one Resource Group per project environment is a good starting point. This enables administrators to control as well as see resource usage at a project environment level. For example, a typical project will have Development, Test and Production environments. A project named ‘CustApp’ would therefore have the following Resource Groups:

- CustApp-Dev-RG
- CustApp-Test-RG
- CustApp-Prod-RG

Access privileges can then be granted to users accordingly. For example, a user whose job role is Developer, would typically have fairly wide-ranging access rights to the development resource group and much tighter or zero access to the production resource group.

Adding Resources to a Resource Group

When creating a resource (remember, a resource is something that is managed using IAM), the name of the Resource Group that it will reside in must be provided. Note that once a resource is created, it cannot change its resource group. If a mistake is made at this stage, the resource must be deleted and recreated, using the correct resource group name. Creating a resource is done via the catalogue and is really no different to creating other services which are not managed by IAM.

What permissions (Access Policies) are required?

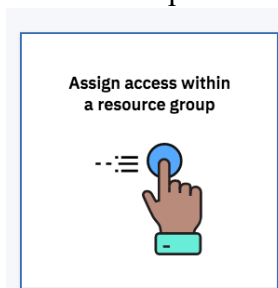
The IBM Cloud account owner can add resources to any Resource Group. Anybody else will need to be granted access rights.

The minimum Platform Management role required to add resources to a resource group is the Editor Role. To assign this:

Click Manage -> Account -> Users. On the user that requires access, click the three horizontal dots to the right of the ‘Status’ column in the table. Select Assign Access from the menu.

User	Email	Status	
James Belton	jamesthecal@gmail.com	ACTIVE	...
JAMES BELTON <small>owner self</small>	james.belton@uk.ibm.com	ACTIVE	Manage user Assign access Remove user
JOHN EASTON	jkj@uk.ibm.com	ACTIVE	

Click the Assign Access within a Resource Group button:



From the drop down, select the name of the Resource Group that access is required for. This will then expose two more drop-downs.

To give minimal access, select ‘Viewer’ in the ‘Assign access to Resource Group’ drop down. In the Services drop down, select ‘All Identity and Access Services’. This will reveal a further drop down and several tick boxes.

In the Region drop down, select the region which the user can affect. Use ‘All Regions’ to allow them to create resources in any region but if you want to restrict them to a particular region, select the appropriate option.

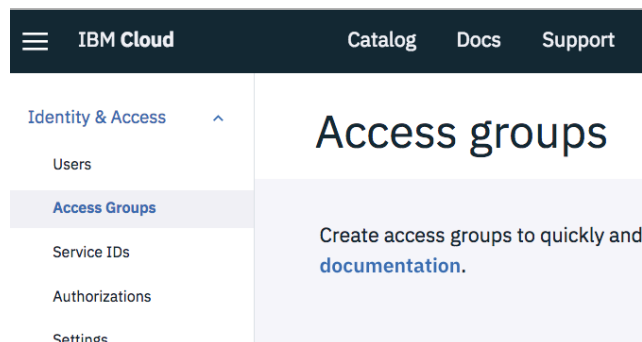
Under ‘Assign platform access roles’, click Editor and then click Assign. This will allow that user to create resources in that resource group.

Using an Access Group

An Access Group is a collection of User IDs and / or Service IDs to which IAM permissions can be granted. Access Groups are useful where there are multiple IDs that need the same access permissions – rather than grant them individually, multiple times, by using an Access Group they can be assigned once to the Group and the appropriate IDs then assigned to the Group.

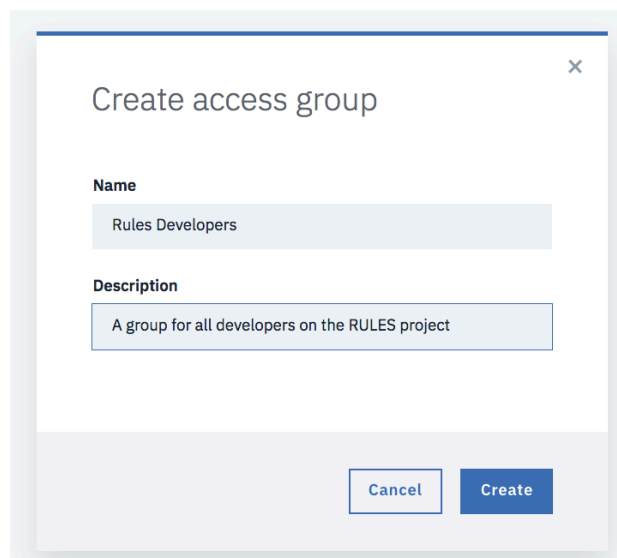
How to Create an Access Group

Creating an Access Group is simple. Click Manage -> Security -> Identity & Access and then Access Groups from the left-hand menu.



This will display any Access Groups which are currently active in the account. By default, there are none.

To create a new Access Group, click the 'Create' button and then provide a name and description of the Access Group in the pop-up box. Then click 'create'.

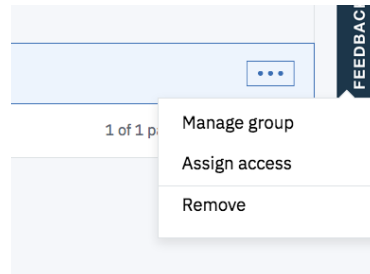


When the Group is created, the screen automatically changes to allow users to be added to it.

Assigning Users and Service IDs to Access Groups

To assign Users and Service IDs to an Access Group, first navigate to the list of Access Groups on the account by clicking Manage -> Security -> Identity & Access and then Access Groups from the left-hand menu.

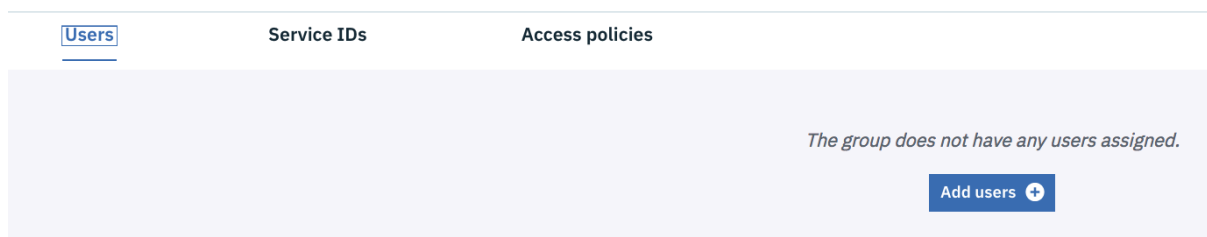
This will display a table showing the Access Groups that are active in the account. Hover the mouse over the row containing the Access Group that requires users assigning and click on the three horizontal dots that appear. Then choose Manage Group from the menu that appears.



The next screen allows Users to be added to the Access Group. Ensure that Users is highlighted and then click the ‘add users’ button.

Manage Rules Developers

ID: AccessGroupId-de03abba-e075-4ad8-b604-8a8182021d50 Description: A group for all developers on the RULES project



The next screen displays a list of users that can be added to the Access Group, with a tick box beside each. Check the box beside the users that need to be added to the Group and then click the ‘Add to Group’ button.



Adding Service IDs to an Access Group

Assigning a Service ID to an Access Group is much the same process as adding a User ID.

Navigate to the list of Access Groups on the account by clicking Manage -> Security -> Identity & Access and then Access Groups from the left-hand menu.

This will display a table showing the Access Groups that are active in the account. Hover the mouse over the row containing the Access Group that requires Service IDs assigning and click on the three horizontal dots that appear. Then choose Manage Group from the menu that appears.

Ensuring that Service IDs is highlighted, click the Add Service ID button, which will then display all of the Service IDs that can be added to the Access Group, with a tick box next to each. Check the box of the services that need to be added to the access group and then click the ‘Add to Group’ Button.

Assign Access Policies to an Access Group

Assigning permissions or Access Policies to an Access Group is much like the process of assigning them to an individual.

For example, in the section ‘Adding Resources to a Resource Group’ above, we discussed the permissions that a developer would need to create a resource into a particular resource group. To assign the same permissions to multiple developers in one hit, simply assign the same permissions to the Access Group.

Navigate to the list of Access Groups on the account by clicking Manage -> Security -> Identity & Access and then Access Groups from the left-hand menu.

This will display a table showing the Access Groups that are active in the account. Hover the mouse over the row containing the Access Group to which permissions need to be assigned and click on the three horizontal dots that appear. Then choose Manage Group from the menu that appears.

Highlight Access Policies and then click Assign Access.

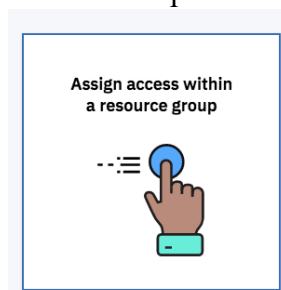
Manage Rules Developers

ID: AccessGroupId-de03abba-e075-4ad8-b604-8a8182021d50

Description: A group for all developers on the RULES project

Users	Service IDs	<u>Access policies</u>
<p><i>The group does not have any access assigned.</i></p> <p>Assign access +</p>		

Click the Assign Access within a Resource Group button:



From the drop down, select the name of the Resource Group that access is required for. This will then expose two more drop-downs.

To give minimal access, select ‘Viewer’ in the ‘Assign access to Resource Group’ drop down. In the Services drop down, select ‘All Identity and Access Services’. This will reveal a further drop down and several tick boxes.

In the Region drop down, select the region which the user can affect. Use ‘All Regions’ to allow them to create resources in any region but if you want to restrict them to a particular region, select the appropriate option.

Under ‘Assign platform access roles’, click Editor and then click Assign. This will allow all users (and / or ServiceIDs) assigned to the Access Group to create resources in the resource group.

Manage Rules Developers

ID: AccessGroupId-de03abba-e075-4ad8-b604-8a8182021d50 Description: A group for all developers on the RULES project

Users Service IDs Access policies

Based on your assigned role, you can click the role to view or edit the policy.

Role ↕	Access Type	Policy Details
Editor	Resource	All resources in RULES-DEV resource group
Viewer	Resource group	Only resource group RULES-DEV itself

How Roles make Policies that define Access

Unless the account administrator, by default no users have any access to any resources. To provide access, they must be granted permissions. These permissions are in the form of Roles and the combination of roles granted becomes a Policy.

For example, we have seen in the sections above, that to enable a developer to create a resource within a resource group, they must be granted the 'Viewer' role on the target Resource Group and then the 'Editor' role on all IAM Resources. Together, these make a Policy, which could be described as 'Provide minimal access to developers to create resources in a particular resource group'.

The Roles provided are fixed but they can be applied in pretty much any combination, so it's possible to create both simple and complex Policies which provide access to either a whole range of resources or to just one particular service. When creating an access policy, getting it just right may even take several iterations.

What are the roles and what do they do?

The roles that can be assigned in IAM are split into two types – Platform Management Roles and Service Access Roles.

Platform Management Roles

These roles, as the name suggests, enable management of the platform itself. Through them, users are granted a number of privileges, including the ability to create instances, manage service IDs, manage users and permissions, and create resource groups. Platform roles are administrator, editor, operator, viewer.

The table below is lifted from the official IBM Cloud documentation and describes example platform management roles and actions.

Access policy details	Actions a user can perform on services in the account	Actions for service IDs	Actions for access to resource groups	Action on resources within resource groups
Assign access to	One or all IAM-enabled services	IAM Identity Service	Selected resource group	Selected service in a resource group
Viewer role	View instances, aliases, bindings, and credentials	View IDs and API keys	View resource group	View only specified instances in the resource group
Operator role	View instances and manage aliases, bindings, and credentials	Not applicable	Not applicable	Not applicable
Editor role	Create, delete, edit, and view instances. Manage aliases, bindings, and credentials	Create and delete IDs and API keys	View and edit name of resource group	Create, delete, edit, suspend, resume, view, and bind only specified instances in the resource group
Administrator role	All management actions for services	Create and delete IDs and API keys, assign policies to IDs	View, edit, and manage access for the resource group	All management actions for the specified instances in the resource group

Viewer Role

The viewer role, if granted, will in general, allow a user to see a resource group and the services that have been created inside it. This can be configured so that they can see all services, or one or more particular services. With this role, the user cannot do anything with the services they can see.

Operator Role

The operator role is not visible in all cases and this is normal. Where the Operator role is present, if granted, the user will be able to use a service instance but not manage or change it in any way.

Editor Role

If granted, the editor role will allow a user to create and change service instances. This can be granted on all resources in a resource group or it can be targeted to a particular resource type or even a single instance.

Administrator Role

This role, if granted, will provide full administrator rights and allow a user to fully manage resources. This can either be granted to all resources or can be tied down to a particular resource type or resource instance. If a user creates a resource, they will have the Administrator role on that resource by default.

Service Access Roles

Service access roles define a user or service’s ability to perform actions on a service instance such as accessing its User Interface or performing API calls. There are three possible service access roles: manager, writer, and reader. Note that each service may vary the way in which it implements these roles. As an example, the following table shows how the Object Storage service implements the roles:

Service access role	Description of actions	Example actions for Object Storage service
Reader	Perform read-only actions within a service such as viewing service-specific resources	List and download objects
Writer	Writers have permissions beyond the reader role, including creating and editing service-specific resources.	Create and destroy buckets and objects
Manager	Managers have permissions beyond the writer role to complete privileged actions as defined by the service. In addition, you can create and edit service-specific resources.	Manage all aspects of data storage, create and destroy buckets and objects

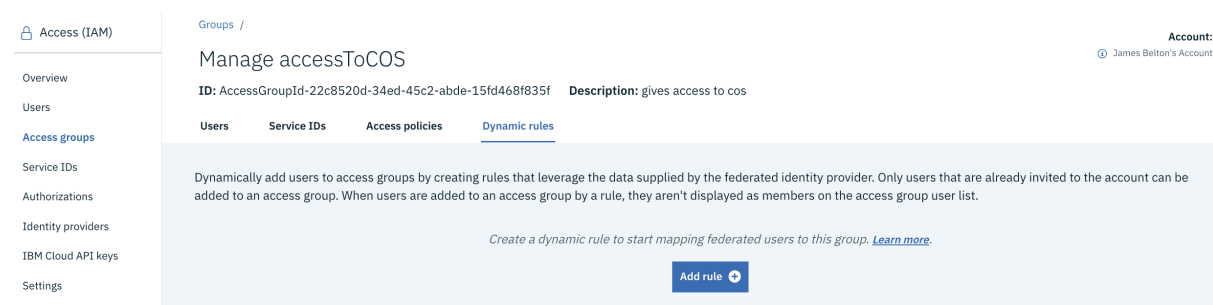
Combining Access Groups with Identity Federation

Many organizations adopt 'single-sign-on' practices with the applications that they use. Typically, this is signing into some kind of identity management system, for example Microsoft Access Directory, and those same credentials are then 'validated' against other systems they use to provide access without a need to have multiple log-ins.

IBMid Identity Federation is a means by which IBM customers can use their own identity providers (e.g. Active Directory) to have a seamless 'single sign on' to IBM Cloud. I won't go into detail on this here but I wrote a short blog on the subject which will help explain the concept and is available at <https://developer.ibm.com/dwblog/2018/identity-federation-ibm-cloud/>, which I recommend you read, particularly if the following paragraphs are

Once an organization has federation in place, they can then use the SAML token which is exchanged as part of the federated identity login process to dynamically assign users to Access Groups in IBM Cloud, based on memberships in their own on-premises directory. In the context of Active Directory, this means that certain group membership in Active Directory can be mapped to access in IBM Cloud. The advantage of this is that group membership in IBM Cloud can be controlled from group membership in an organization's Active Directory system, meaning there is a single place for such group and user administration (the Active Directory system).

To enable dynamic assignment to Access Groups, aside having Identity Federation in place, Dynamic Rules need to be created for the Access Group(s). To create a Dynamic Rule, first locate the Access Group (Manage -> Access (IAM) -> Access Groups -> Select group) and then click Dynamic Rules.



To create a rule, click Add rule.

In the top half of the screen, you will see a section headed 'Identity provider data'. This is the attribute data that is being passed in the SAML token from your identity provider and is intended to assist in creating the rules.

In the lower part of the screen, you create your rule.

The screenshot shows a form for creating a new rule. It has three main sections: 'Name', 'Identity provider', and 'Expiration (in hours)'. The 'Name' field is empty with a placeholder 'Name of the rule'. The 'Identity provider' field is empty with a placeholder 'Identity Provider for the rule' and an example URL. The 'Expiration (in hours)' field is a dropdown menu set to '1'. Below these are 'Add users when', 'Comparator', and 'Values' sections. 'Add users when' is empty with a placeholder 'Identity provider attribute name' and examples. 'Comparator' is a dropdown menu set to 'Select comparator'. 'Values' is empty with a placeholder 'Identity provider attribute value' and examples.

A quick run through of the fields and what they are.

Name: Provide a meaningful name for your rule, for example ‘cloud-developer’.

Identity Provider: This is the URL of the identity provider, for example, the public URL address of your organisation’s Active Directory Federated Services service

Expiration in Hours: this is how long you want access to the access group to be provided, up to 24 hours. This can be useful where you want to provide time-limited access to resources.

Add users when: this is the name of an attribute within the token that is exchanged. Examples of the attributes which are passed in the token are shown below the field.

Comparator: this is set to comparisons such as ‘equals’, ‘less than’, ‘greater than’, ‘contains’ etc.

Values: This is the value of the attribute to be evaluated

For example, a rule might look at an attribute called ‘blueGroups’ and if that attribute contains the value ‘cloud-developer’, it will dynamically assign the access group to the user at login for 5 hours. To set this rule up, the screen might look as follows:

The screenshot shows the same form as before, but with example values filled in. The 'Name' field contains 'cloud-developer-access-group-dyn-rule'. The 'Identity provider' field contains the URL 'https://w3id.sso.ibm.com/auth/sps/samlidp2/saml20'. The 'Expiration (in hours)' field is set to '5'. The 'Add users when' field contains 'blueGroups'. The 'Comparator' field is set to 'Contains'. The 'Values' field contains 'cloud-developer'.

You can also add further conditions as AND arguments (e.g. AND ‘company’ equals ‘IBM’).

To create and assign the rule, click the Add Rule button.

Once the rule is created, when any account user logs in, the rule will run and evaluate whether to dynamically add the user to the Access Group and there is no longer a need for the user to be added to the group via the IBM Cloud console. If the token received does not meet the conditions for the rule, the user will not have access to the Access Group.

Two Examples

Below I set out two example scenarios and how they can be implemented using IAM. Remember that in many cases, it may be simpler to assign access via an Access Group, rather than directly to a user or Service ID.

Scenario 1 – Developer Access to an Object Storage Bucket.

This is quite a common example, where a developer (or it could be a service ID) needs access to a bucket which has been created via Object Storage. Here, access needs to be granted so that the developer can access one particular bucket only.

The name of the Object Storage instance is ObjectStorage-RulesDev and the bucket name is rules-dev-forms-in. The object storage instance resides in a resource group called Rules-DEV.

The first step is to assign the user Viewer level access on the resource group. To do this, click Manage -> Account -> Users and from the table, click on the user that needs the access.

On the next screen, ensuring that Access Policies is highlighted, click Assign Access.

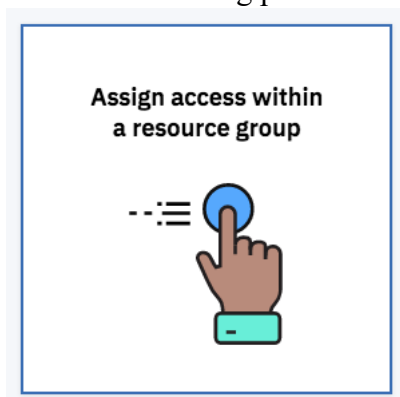
Manage James Belton ④ ● ACTIVE

[Access policies](#)[Cloud Foundry access](#)[Access group](#)

This user does not have any access assigned.

[Assign access](#) +

Click the following panel:



On the next screen, select the name of the resource group (in this case Rules-DEV) from the resource group list and select the 'Viewer' Role. From the services drop down, select Cloud Object Storage and then leave Resource Type blank but check the Viewer platform access role and the Writer service Access role then click Assign.

IBM Cloud Identity and Access Management – A beginner’s guide to understanding IAM

Assign resource group and resource access to James Belton ① ● ACTIVE

James

When you assign a user access to a resource within a resource group, you can also assign access for viewing, editing, or managing access to the group. Select a role for the Assign access to a resource group option to provide access to the group. Select "No access" if you want to only provide access to a specified resource.

Resource group
RULES-DEV

Assign access to a resource group ①
Viewer
As a viewer, you can view service instances, but you can't modify them.

Services
Cloud Object Storage

Resource type

Select roles

Assign platform access roles
These roles enable users to complete tasks on platform resources, such as creating instances, connecting instances to apps, and assigning user permissions.

- ☐ Administrator
As an administrator, you can perform all platform actions based on the resource this role is being assigned, including assigning access policies to other users.
- ☐ Editor
As an editor, you can perform all platform actions except for managing the account and assigning access policies.
- ☐ Operator
As an operator, you can perform platform actions required to configure and operate service instances, such as viewing a service's dashboard.
- ☒ **Viewer**
As a viewer, you can view service instances, but you can't modify them.

Assign service access roles
These roles allow a user access for using the service and performing service API calls.

- ☐ Manager
As a manager, you have permissions beyond the writer role to complete privileged actions as defined by the service. In addition, you service-specific resources.
- ☒ **Writer**
As a writer, you have permissions beyond the reader role, including creating and editing service-specific resources.
- ☐ Reader
As a reader, you can perform read-only actions within a service such as viewing service-specific resources.

This will take you back to the user screen, where you should see a table similar to the following:

Manage James Belton ① ● ACTIVE

Access policies

Cloud Foundry access

Access group

Based on your assigned role, you can click the role to view or edit the policy.

Role	Access Type	Policy Details
Viewer	Resource group	Only resource group RULES-DEV itself
Viewer, Writer	Service	All Cloud Object Storage resources in RULES-DEV resource group

This confirms that the Viewer role has been granted on the RULES-DEV resource group and that the user is able to view and write for cloud object storage resources. However, this doesn't explicitly give access to the bucket. The next step is to provide access so that the user can write to a bucket created within the service.

This is actually most easily managed from the Object Storage screens. So, from the dashboard, use the drop downs to show the resources in the Rules-DEV resource group:

Dashboard

RESOURCE GROUP
RULES-DEV

CLOUD FOUNDRY ORG
None

CLOUD FOUNDRY SPACE
All Spaces

LOCATION
All Locations

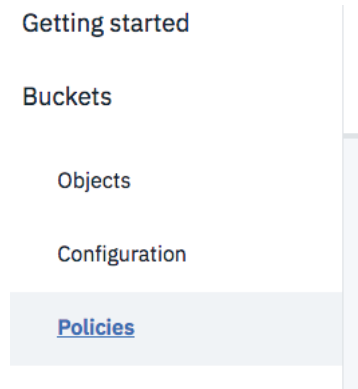
CATEGORY
All Categories

Services

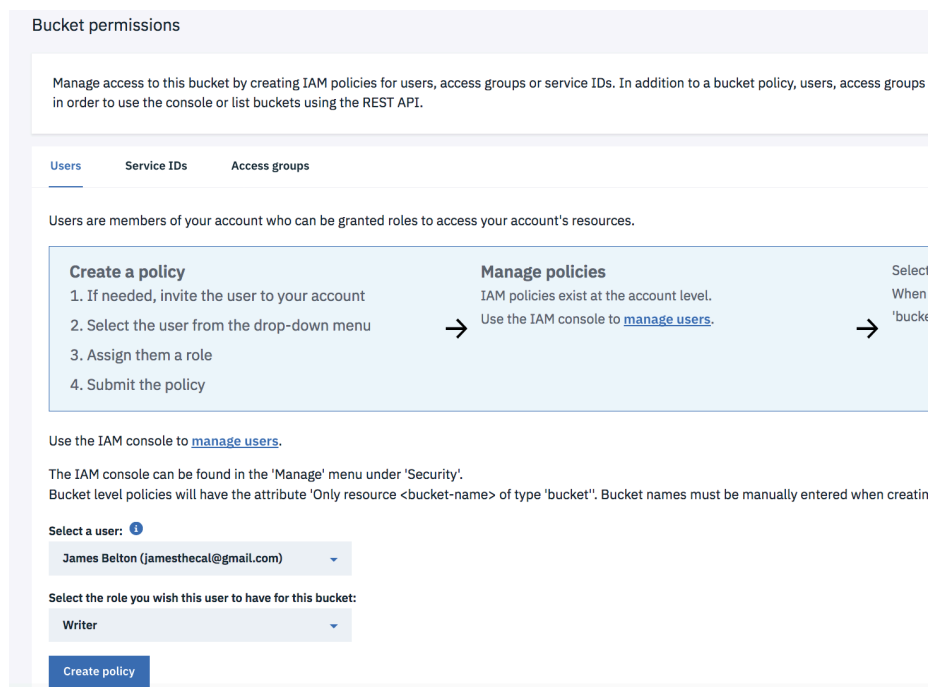
Name	Location
ObjectStorage-RulesDev	global

Click on the name of the Object Storage instance.

On the next screen, click Buckets from the left-hand menu and then click on the name of the bucket to which access is to be given. This will expand the area under Buckets in the left-hand menu; click on Policies.

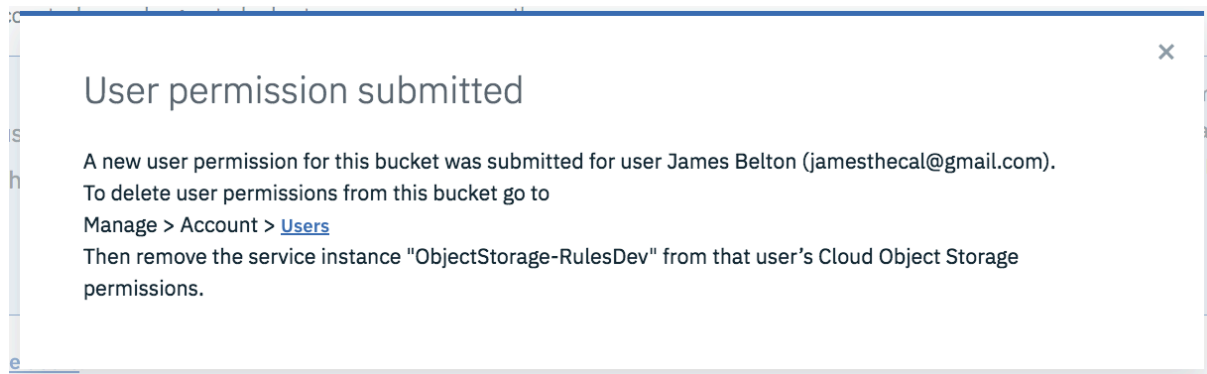


The main part of the screen will then change to allow access policies to be defined, as per the screen below.



Since we’re giving access directly to a user, Users is highlighted at the top and then use the first drop down to select the user that needs access. If the user you want to give access to is not shown in the drop down, then ensure the user has been invited to the account. Next, set the role you want to assign, which gives the level of access you need. Here, ‘Writer’ will allow the user to write to the bucket. The other options are ‘Reader’ which will allow the user to retrieve and read documents from the bucket and ‘Manager’ which will allow the user to read, write and manage the bucket. Here, we just want to assign the Writer role, so then click ‘Create Policy’.

The following popup is displayed to confirm the action:



The user will now be able to view and use the bucket from their account.

Scenario 2: Three Teams working on Three Different Projects.

This is another common example which is typical on larger accounts where there are several separate projects happening at once. In this example, we'll have three users who will be administrators that need to set things up and keep things going at an account level and then there are a total of nine developers, who work in teams of three across the three projects.

The Administrators

The three administrator users need to be able to add new users to the account, create new resource groups and assign access to the resource groups the users. If the administrators do not yet exist on the account, then they must first be invited, at which point the permission can be set. To do this:

1. from the Console click Manage -> Security -> Identity & Access. Ensuring that Users is highlighted in the left-hand menu, next click the 'Invite Users' button.
2. On the next page, enter the email addresses of the users that you want to become administrators. Note that you can do this one user at a time or all three at once.
3. Next, click on Services to expose the drop-downs in this section. Leave 'Assign Access To' as *Resource* and set the 'Services' drop-down to *All Identity and Access Services*.
4. Ensure that 'Region' is set to *All Regions* and the check the *Administrator* box under 'Assign platform access roles'
5. Click Invite Users

If the administrators already exist on the account, then the route through is slightly different.

1. From the Console, click Manage -> Security -> Identity & Access. This will display a list of users. From that list hover over the user that you want to make an administrator and click the three horizontal dots that appear on the right-hand side to reveal a menu.
2. Click *Assign Access* from the menu and then *Assign Access to Resources* from the following screen.
3. From the 'Services' drop-down, select *All Identity and Access enabled services* and ensure that *All Regions* is selected in the 'Region' drop-down.

4. Then, check *Administrator* from the ‘Assign Platform Access Roles’ panel and click the ‘Assign’ button. Repeat for the other administrators.

Having completed these steps, the users will have administrator level rights on the account and be able to create resource groups in any region as well as invite users and assign rights to those users.

In this example, there are three separate projects. The best practice advice in this situation is to create a separate resource group for each of the projects. This provides the separation needed and allows administrators to view costs split by resource group and therefore project. Each resource group should be named in a way that allows for easy identification (e.g. *ProjectName_EnvironmentName*) so in this example, the Administrator(s) will now create three resource groups called:

- *ProjectA_DEV*
- *ProjectB_DEV*
- *ProjectC_DEV*

The Developers

In this example, there are a total of nine developers, split evenly across the projects. So that’s three developers for each project. These users only need to access resources within their respective project resource groups. Assuming these users do not already exist in the account, we will need to invite them and this can be done in three groups as follows:

1. from the Console click Manage -> Security -> Identity & Access. Ensuring that Users is highlighted in the left-hand menu, next click the ‘Invite Users’ button.
2. On the next page, enter the email addresses of the users that you want to become developers in resource group *ProjectA_DEV*.
3. Next, click on Services to expose the drop-downs in this section. Change ‘Assign Access To’ as *Resource Group* and change the ‘Resource Group’ drop down to *ProjectA_DEV*.
4. In the ‘Assign Access to a Resource Group’ drop-down, select the appropriate access right. Select *Editor* where a user needs to create resources (i.e. service instances) and select *Viewer* to allow users to see what resources there are but they will be unable to create new ones or modify existing services.
5. The next set of drop-downs allows us to get very granular and set access rights on particular services but at this point, we’ll grant users access to all services. So, select *All Identity and Access enabled Services* from the ‘Services’ drop-down and *All regions* from the ‘Regions’ drop down.
6. Next, check *Editor* and *Manager* where the user requires the ability to create services in the resource group and manage them as well as access to them or, check *Viewer*, and *Reader*, *Writer* for a user that just needs to use services but cannot create new ones or assign access to them.
7. Repeat for the other resource groups and users.

Where users already exist:

1. From the Console, click *Mange* -> *Securtiy* -> *Identity & Access*. This will display a list of users. From that list hover over the user that you want to make an administrator and click the three horizontal dots that appear on the right-hand side to reveal a menu.
2. Click *Assign Access* from the menu and then *Assign Access Within a Resource Group* from the following screen.
3. Choose the correct resource group from the ‘Resource’ drop-down menu.
4. The next set of drop-downs allows us to get very granular and set access rights on particular services but at this point, we’ll grant users access to all services. So, select *All Identity and Access enabled Services* from the ‘Services’ drop-down and *All regions* from the ‘Regions’ drop down.
5. Next, check *Editor* and *Manager* where the user requires the ability to create services in the resource group and manage them as well as access to them or, check *Viewer*, and *Reader*, *Writer* for a user that just needs to use services but cannot create new ones or assign access to them.
6. Repeat for the other resource groups and users.

The account should now be good to go, with a set of administrators that have oversight over the whole account and three groups of developers, working on separated projects.

Try Experimenting!

As can be seen from the steps, assigning rights can get very granular, so it’s worth experimenting to get the access rights that are best for an individual project.

Again, access can also be grated using Access Groups. If there are large numbers of users who need the same access rights, from an administration point of view, assigning rights via an Access Group may well be easier and less complex than needing to manage access via lots of different users.

Remember that a user can have different access rights to different resource groups and can of course be a member of multiple access groups.