Azure IAM Concepts

who (identity) has *what access* (role) for *which resource*

Is made up of

Scope
(Assignable Scopes)
Where role is available

⚠Permissions should be controlled via Role Assignments not Roles Definition⚠

Scope

Control Plane
(Actions)
~5800 actions exist

Collection of permissions

Data Plane
(Data Actions)
~xx data actions exist

Collection of permissions

⚠Actions - (minus) NotActions = Effective control plane permissions. However, this is not a DENY rule. If another role allows the action the action is allow. Additive RBAC ⚠

Role Definitions (Role)

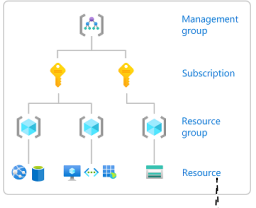
```
{
  "properties": {
    "roleName": "Julians Custom Role",
    "description": "All storage control plane action and all compute except poweroff and
    "assignableScopes": [                                  data access for blob storage",
      "/subscriptions/3aad1bdc-3663-406d-bcb2-c827d4efec...
    ],
    "permissions": [
        {
            "actions": [
                "Microsoft.Compute/*",
                "Microsoft.Storage/*"
            ],
            "notActions": [
                "Microsoft.Compute/virtualMachineScaleSets/powerOff/action",
                "Microsoft.Compute/virtualMachines/powerOff/action"
            ],
            "dataActions": [
                "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read"
            ],
            "notDataActions": []
        }
```

Define permissions and scope of this definition

Contains reference to

You can specify a **scope** at four levels from broad to narrow: management group, subscription, resource group, and resource

3 Scope



⚠Same but not the same⚠

Scope of Role Assignment

Scope
Scope of what identity can access

⚠Use the smallest scope that you need to meet your requirements.⚠

⚠Careful if you assign at the resource level. Harder to manage and limits on number of role assignments⚠

Role Assignments

```
{
  "canDelegate": null,
  "condition": null,
  "conditionVersion": null,
  "description": null,
  "id": "/subscriptions/11111111-1111-1111-1111-111111111111/provid
  "name": "00000000-0000-0000-0000-000000000000",
  "principalId": "22222222-2222-2222-2222-222222222222",
  "principalName": "user@contoso.com",
  "principalType": "User",
  "roleDefinitionId": "/subscriptions/11111111-1111-1111-1111-11111
  "roleDefinitionName": "Contributor",
  "scope": "/subscriptions/11111111-1111-1111-1111-111111111111",
  "type": "Microsoft.Authorization/roleAssignments"
}
```

'Allow' Assignments.
Who (principal) has permissions (Role Definition) to what resources (scope).
An 'additive' role assignment system.

Can Contain

Conditions (Azure ABAC)

Defines access based on attributes associated with security principals, resources, and environment. E.g. allow read access to only resources with 'tag' X or name Y etc.

⚠Only for Storage Blob and Storage Queues Service⚠

🔴Permissions Roles🔴
Owner - Full access to all resources including the right to delegate access to others.
Contributor - Can create and manage all types of Azure resources but can't grant access.
User Access Administrator - Lets you manage user access to Azure resources.

⚠Scope is always "/" (all)⚠

Contain a

Build-in
Created & Maintained
~350 Roles

⚠Not least privilige⚠

Custom

Types

Roles

Manages

Azure RBAC

Authorization. RBAC is an authorization system built on Azure Resource Manager

Uses

View Role assignments for User

Manages

Contains reference to

Deny Assignments

Takes precedence!
(Overwrites allow Role Assignments)

Only available via Azure Blueprints which is pre-release!
Deny assignments block users from performing specific actions even if a role assignment grants them access.
Hence no direct link to Azure RBAC

Azure Key Vault
⚠Separate access policy for data plane⚠

We are not the same thing

Trust Relationship at Subscription level

Azure Active Directory
Identity Access Manager

An Identity Object store.
Full and complex IdP
Responsible for **access/authentication.**
Complex federation possibilities.

🔴Its own product and its own roles assignment system🔴

Tenant

Contains

Azure AD Directory

Contains

Principals
(Security Principal)

Type

Workload Identity
Application

An abstract entity, or template, defined by its application object. The application object is the *global* representation of your application for use across all tenants. Can use SAML or OpenID Connect (AAD)

🔴Complex and multiple security challenges need to be considered here🔴

🔴Humans can easily 'use these' and assume their permissions🔴

Types

Local representation (or application instance) of a global application object in a specific tenant. This object defines what the App can actually do *Whose tokens can be used to authenticate and grant access to specific Azure resources from a user-app, service or automation tool.*

Service Principal
Enterprise Application

'same but 'managed'.

Managed Idenity
No need to manage credentials

Always linked to an Azure Resource, not to an application or 3rd party connector. No one knows the credentials

🔴Key known to the creator of the Service Principal and Consumer
Valid for XX months -> apps break🔴

Service Principal Key
(cert or key)

Your secret (🔑) you need to manage🔑

🔴Humans can authenticate with these keys🔴

Type

System Assigned

Identity is linked to a single Azure Resource. Fully automated. E.g. enable for a VM/Compute instance

Type

User Assigned

Create manual first. Can be linked to multiple Azure Resources

Conditional Access Policies

Humans and Workload Identities (beta) can utilize this. E.g. restrict auth from certain IP addresses or if certain conditions are meet e.g. device risk, risky behavior, MFA (humans)

🔴Role assignments are transitive for groups also. If user A is a member of group B and group B is a member of group C with its own role assignment, user A gets the permissions in group C's role assignment.🔴

Can use

Identity
Humans (not Azure Term)

🔴Groups can contain Groups🔴

Types

Type

User

Guest Users

Group
Contains Users

⚠Use Groups to assign roles not individual users⚠

⚠Resource owner and group owner should be same to avoid owner adding wrong users⚠

⚠Resource owner and group owner should be same to avoid owner adding wrong users⚠

Notes:
Terminology used is official Microsoft terminology.
For Azure RBAC and Azure Active Directory see the official documentation from learn.microsoft.com
This is a visual representation to understand the official documentation with a focus on Azure RBAC not AAD
Strongly recommend testing & trying these concepts 'in Azure' so it 'clicks'.
This covers a lot of key concepts, but external IdP integration/federation and Workload Identity are complex topics
This is not an exhaustive list of security tips/risks around IAM!

Created by Julian Wiegmann - November 2022 V1
Creative Commons License. Contact me for questions
I tried to make it not too complicated nor make mistakes