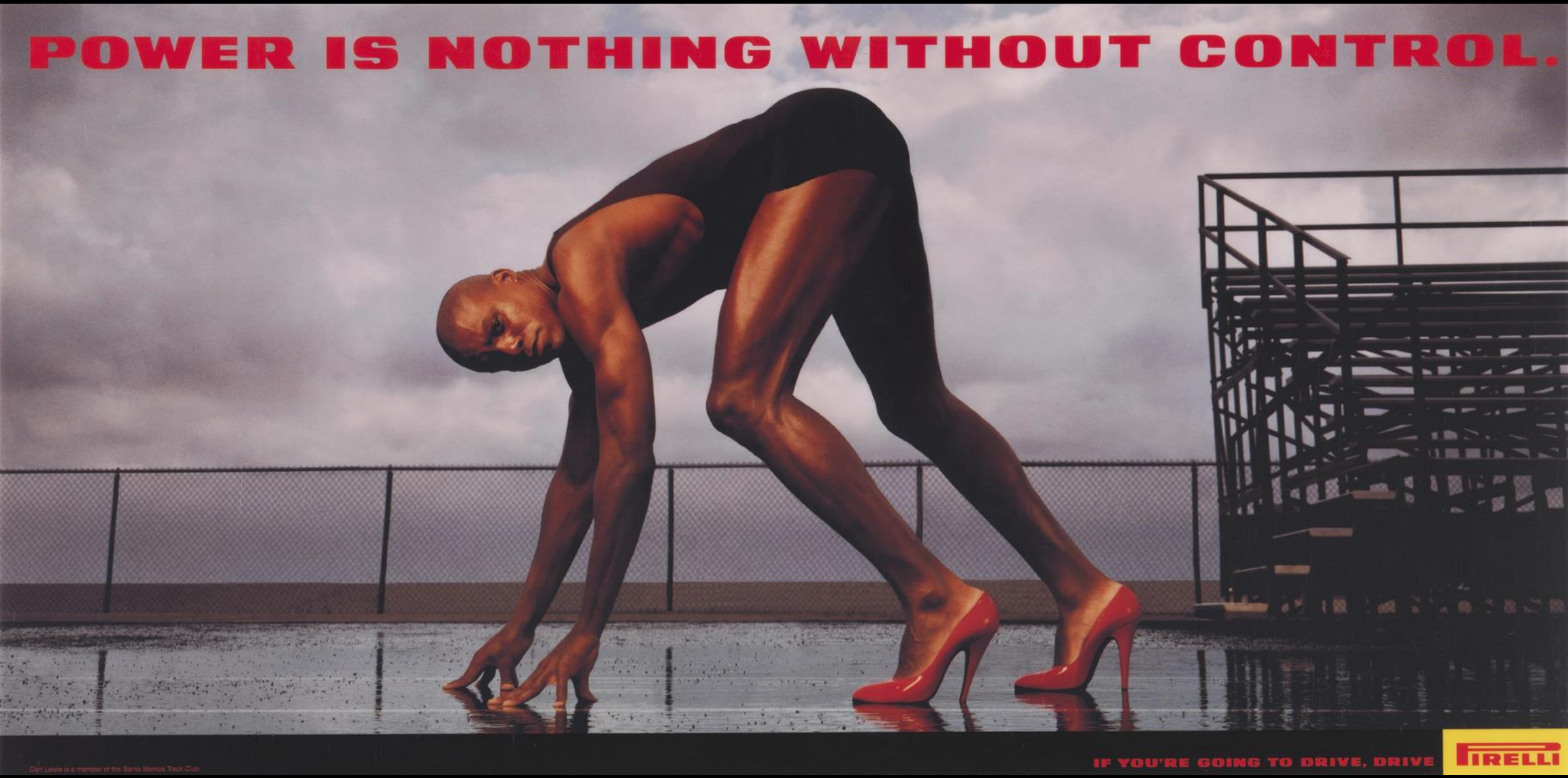
Azure
Meetup

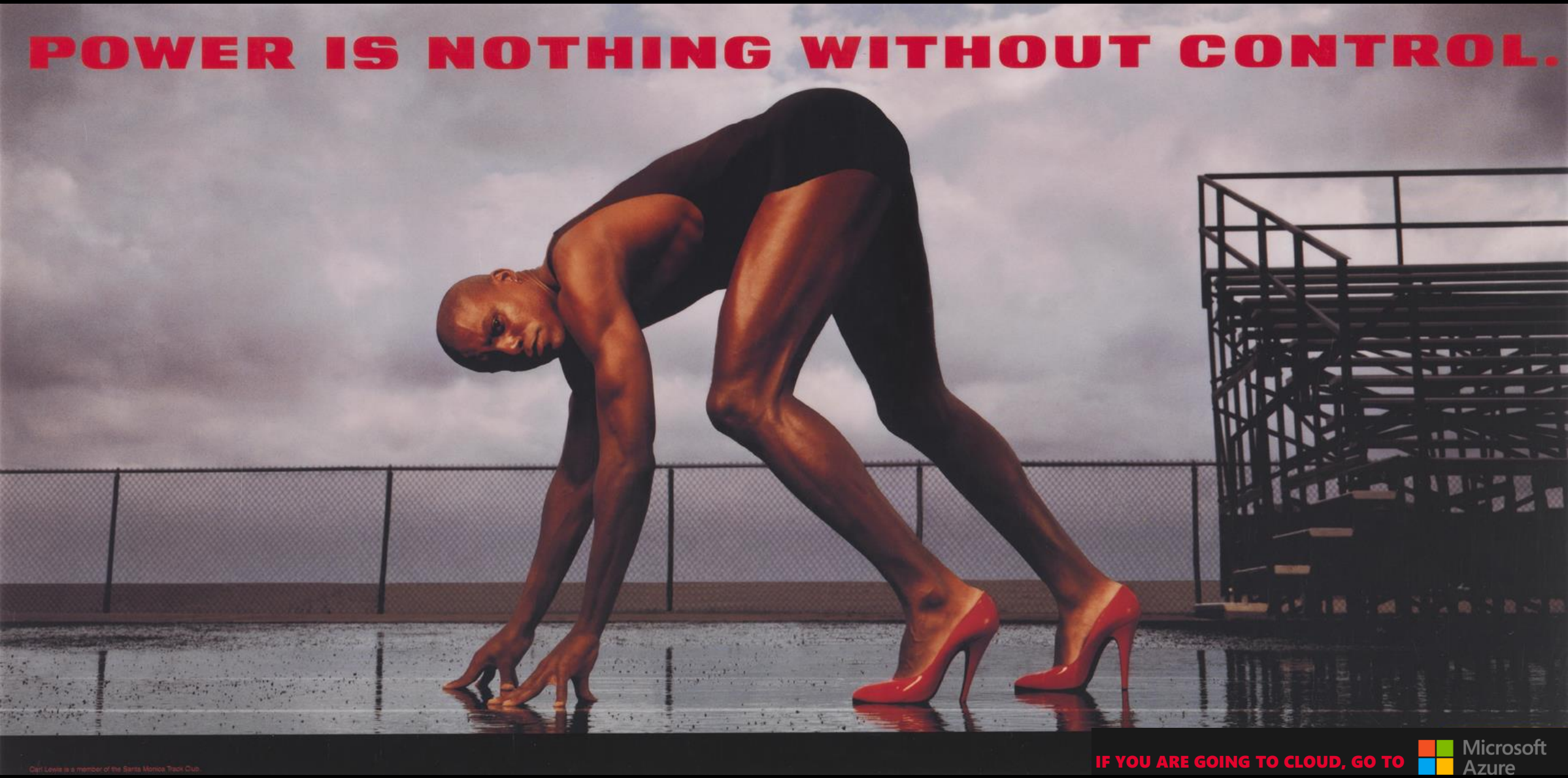
Azure Governance for dummies

Massimo Bonanni

Azure Technical Trainer

massimo.bonanni@microsoft.com
@massimobonanni


POWER IS NOTHING WITHOUT CONTROL.
Carl Lewis is a member of the Santa Monica Track Club.
IF YOU'RE GOING TO DRIVE, DRIVE
PIRELLI


POWER IS NOTHING WITHOUT CONTROL.
Carl Lewis is a member of the Santa Monica Track Club.
IF YOU ARE GOING TO CLOUD, GO TO
Microsoft Azure

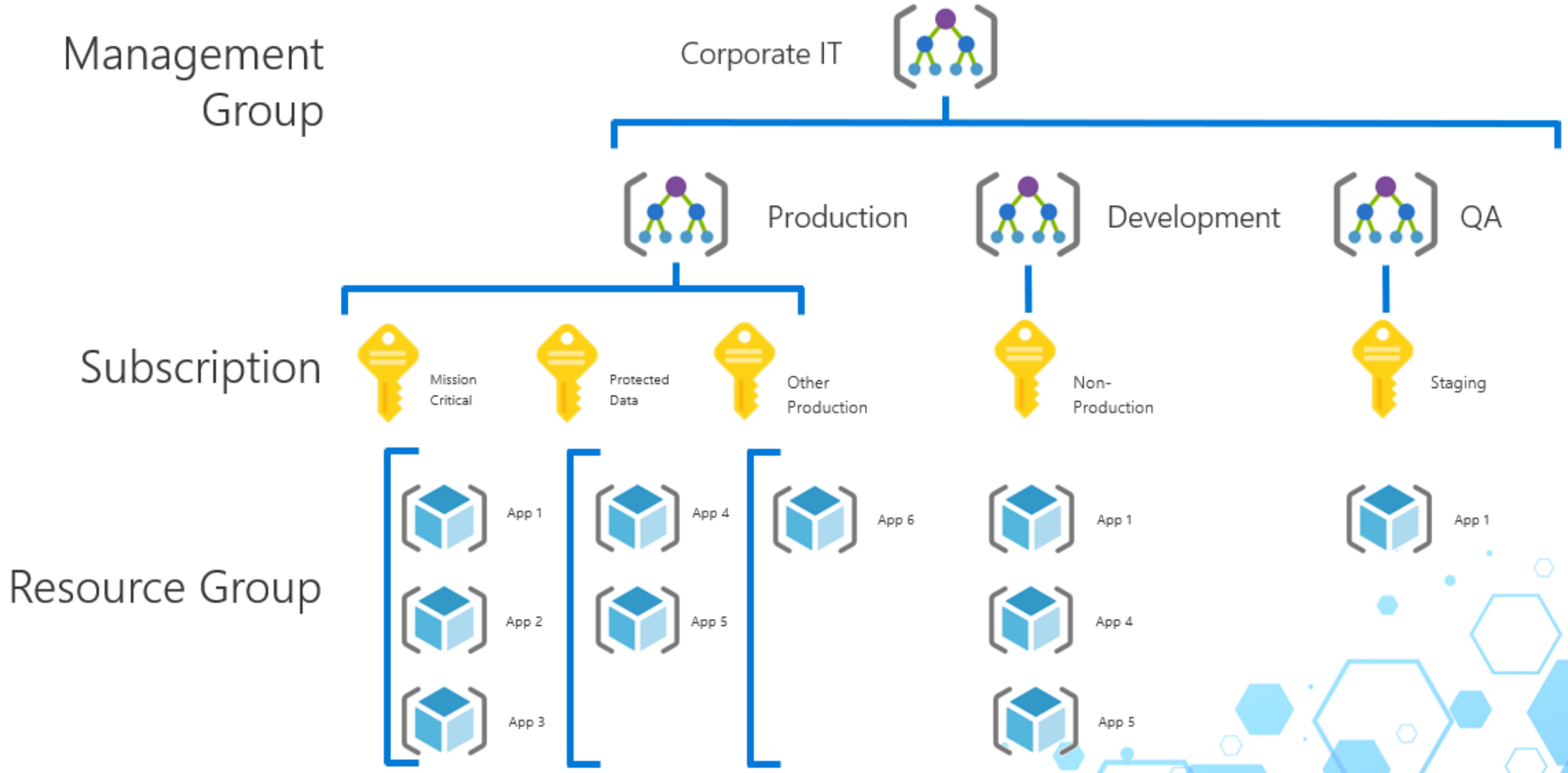
Azure
Meetup

Governance provides mechanisms and processes to maintain control over your applications and resources in Azure.

It involves planning your initiatives and setting strategic priorities

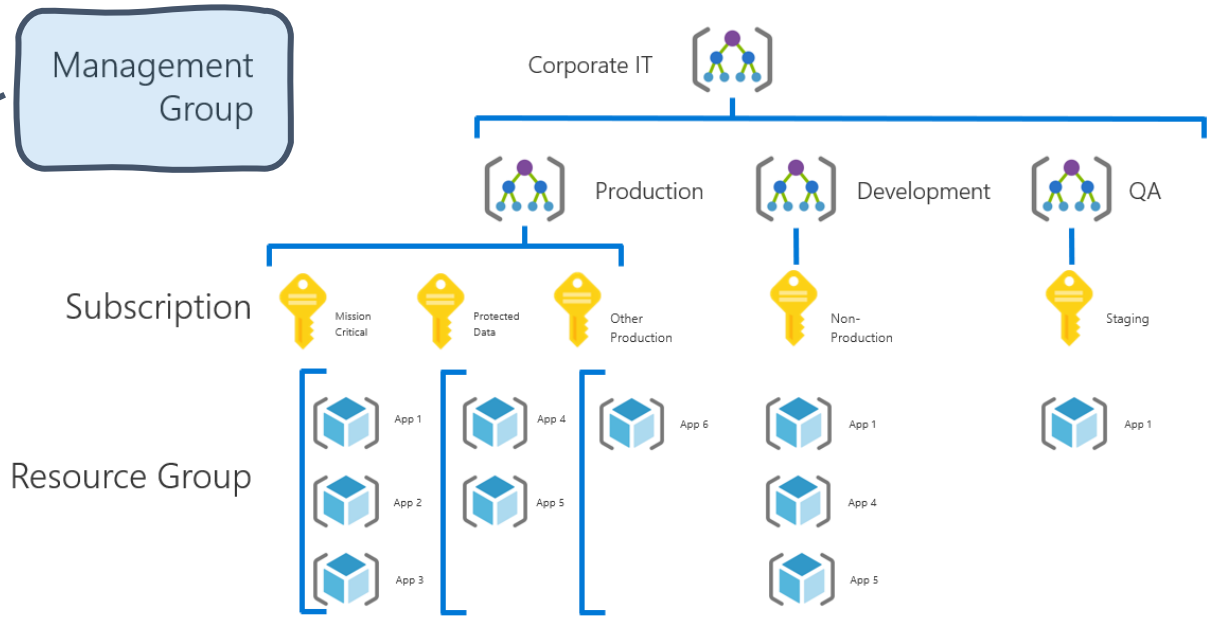
Azure Meetup

Management objects



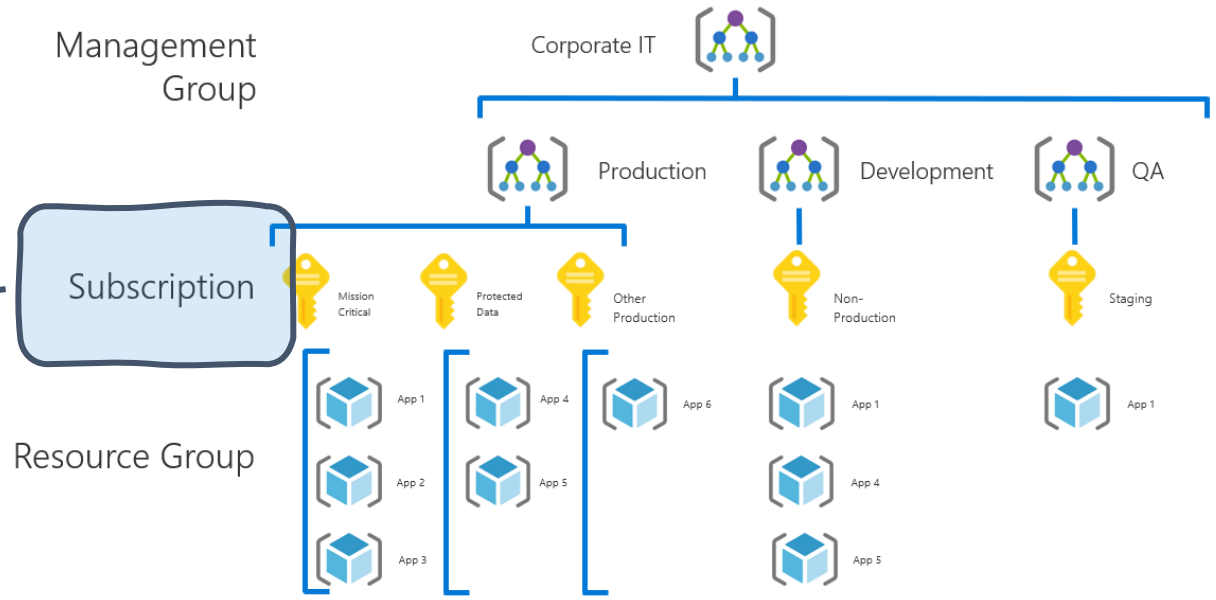
Management objects

- ✓ Azure management groups provide a level of scope above subscriptions.
- ✓ You organize subscriptions into containers called "management groups" and apply your governance conditions to the management groups.
- ✓ All subscriptions within a management group automatically inherit the conditions applied to the management group.
- ✓ Management groups give you enterprise-grade management at a large scale no matter what type of subscriptions you might have



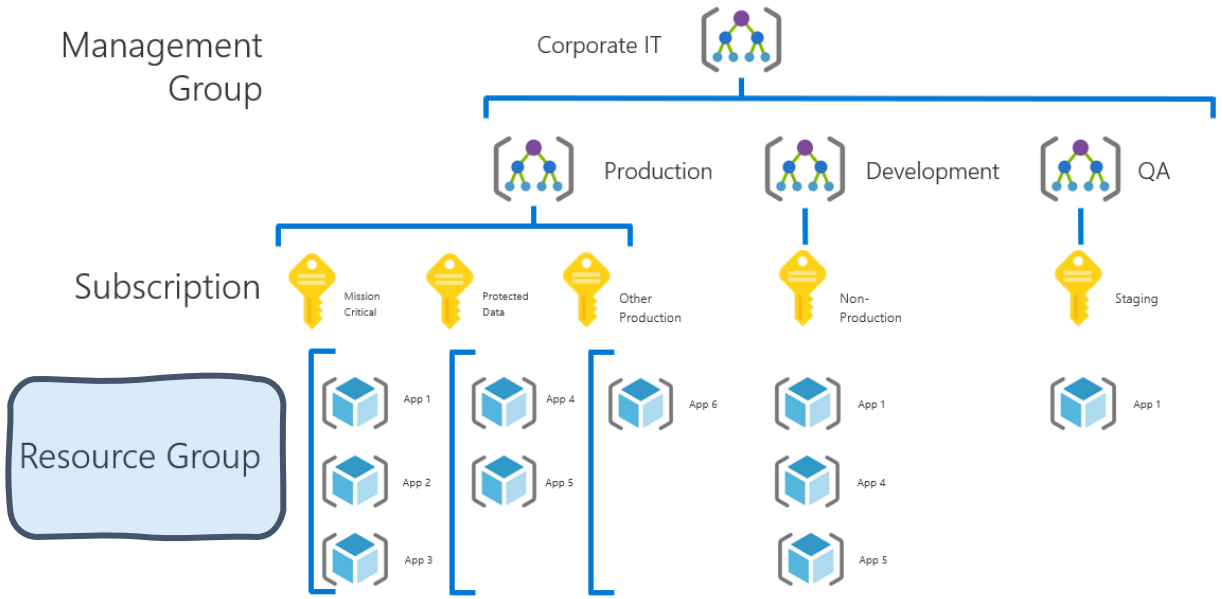
Management objects

- ✓ A subscription is a logical entity that provides entitlement to deploy and consume Azure resources.
- ✓ It is a billing unit.
- ✓ It is a security/policy unit.
- ✓ Depending on the type, these can be free subscriptions, Pay-As-You-Go (Post-Paid) subscription or a pre-paid credit carrying subscription.



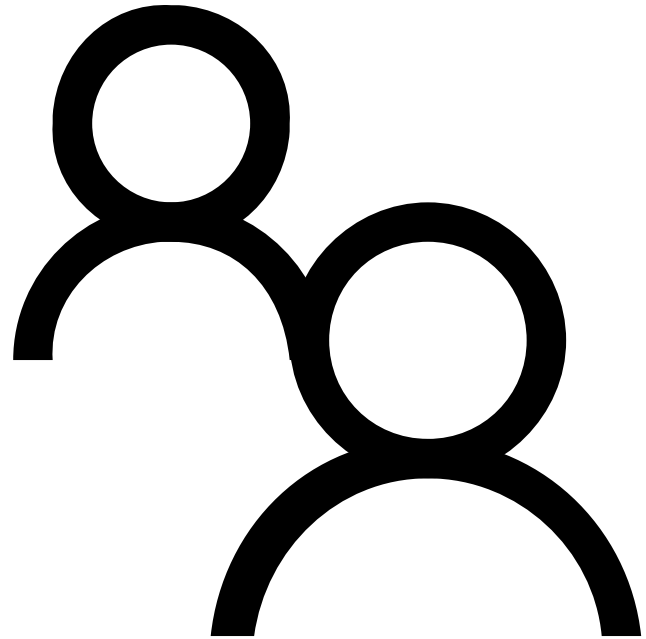
Management objects

- ✓ A resource group is a container that holds related resources for an Azure solution.
- ✓ The resource group can include resources of different types and located in different regions.
- ✓ Generally, the resources in a Resource Group share the same lifecycle.
- ✓ The resource group stores metadata about the resources. Therefore, when you specify a location for the resource group, you are specifying where that metadata is stored.



Azure
Meetup

Role Base Access Control (RBAC)



Azure
Meetup

Azure RBAC is an authorization system built on Azure Resource Manager that provides fine-grained access management of Azure resources.

Azure
Meetup

What can I do with Azure RBAC?

Allow one user to manage virtual machines in a subscription and another user to manage virtual networks

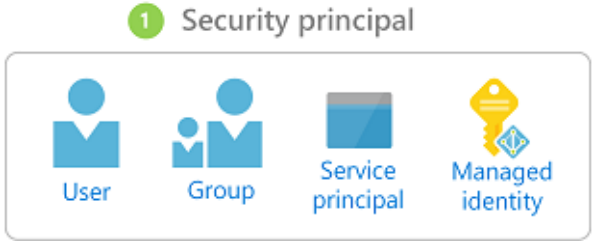
Allow a DBA group to manage SQL databases in a subscription

Allow a user to manage all resources in a resource group, such as virtual machines, websites, and subnets

Allow an application to access all resources in a resource group

Azure Meetup

How Azure RBAC works – Security Principal



User	An individual who has a profile in Azure Active Directory.
Group	A set of users created in Azure Active Directory.
Service principal	A security identity used by applications or services to access specific Azure resources.
Managed identity	An identity in Azure Active Directory that is automatically managed by Azure

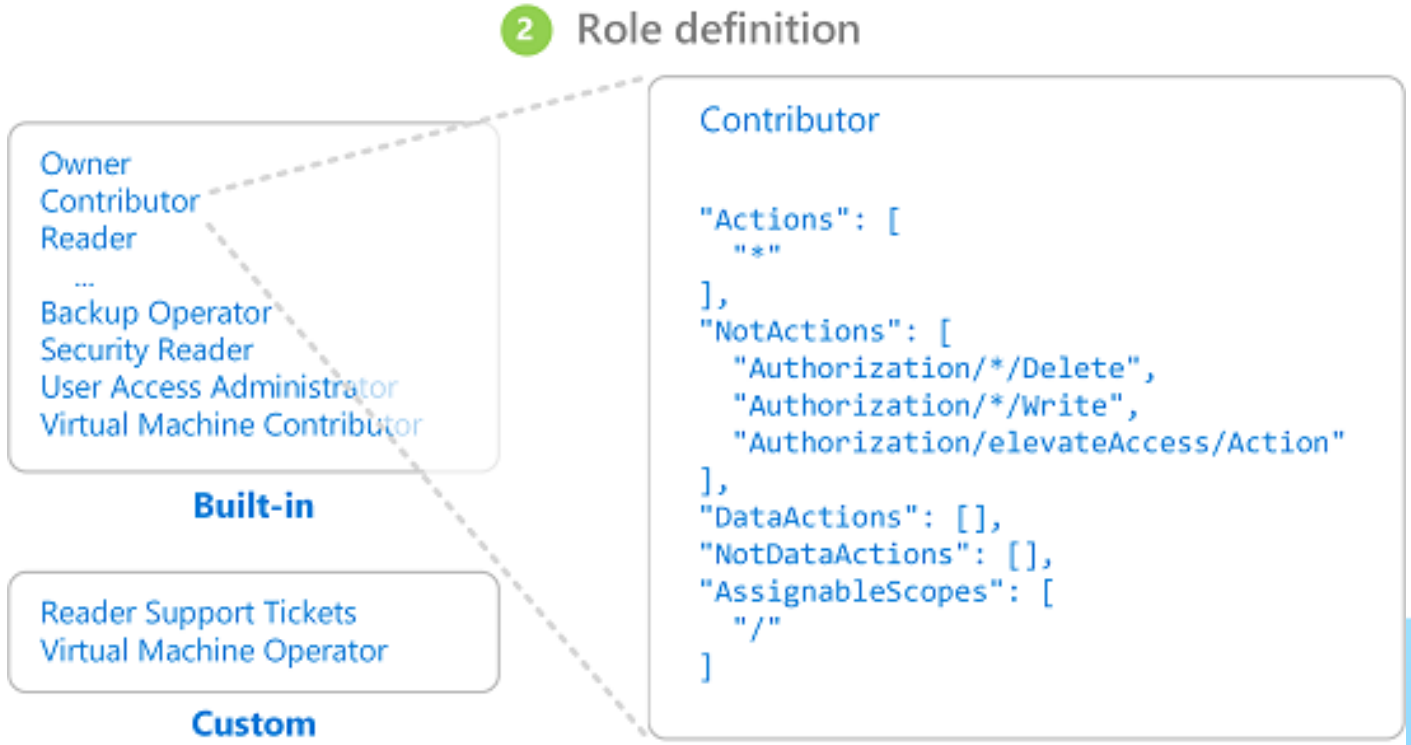
How Azure RBAC works – Role Definition

A role definition is a collection of permissions.

A role definition lists the operations that can be performed, such as read, write, and delete.

Roles can be high-level, like owner, or specific, like virtual machine reader.

Azure includes several built-in roles that you can use.



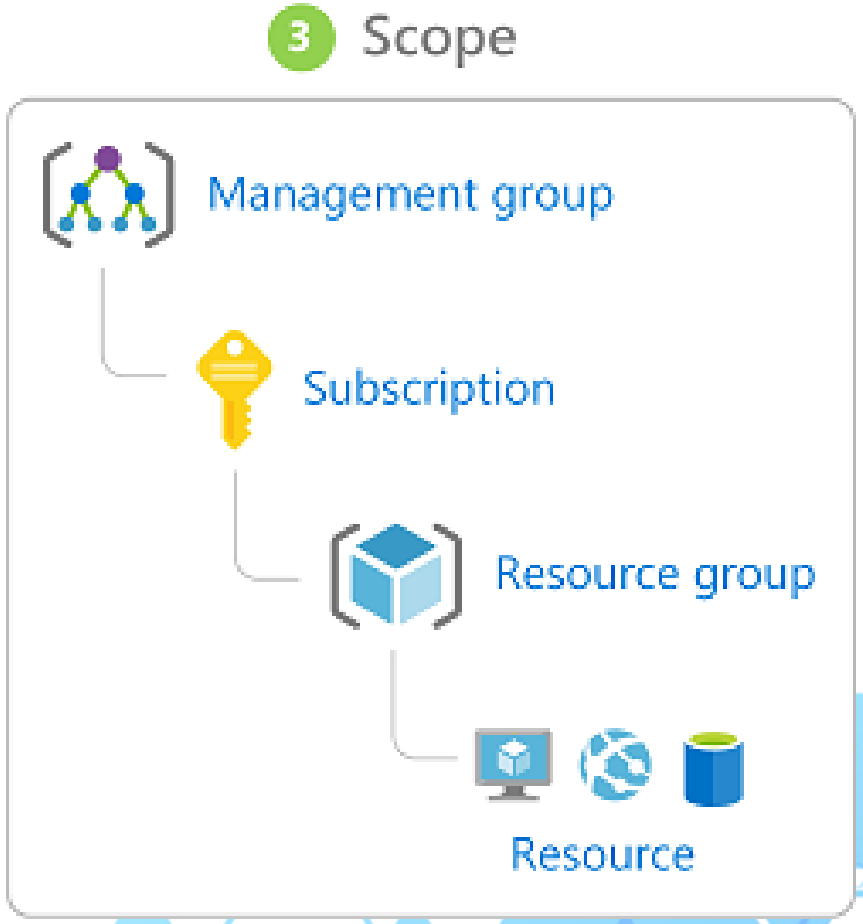
How Azure RBAC works – Scope

Scope is the set of resources that the access applies to.

When you assign a role, you can further limit the actions allowed by defining a scope.

In Azure, you can specify a scope at multiple levels: management group, subscription, resource group, or resource.

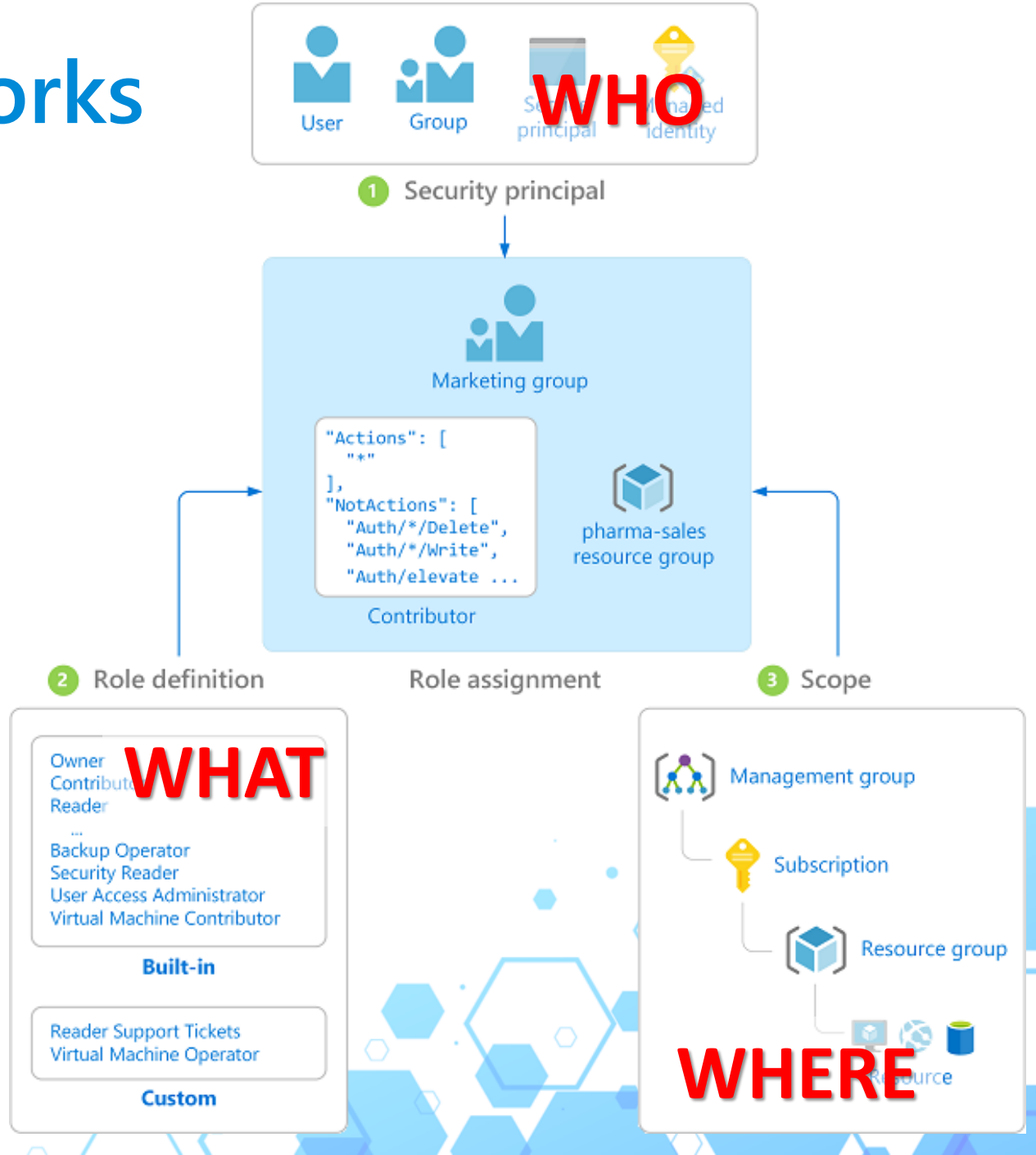
Scopes are structured in a parent-child relationship.



How Azure RBAC works Role assignments

A role assignment is the process of attaching a role definition to a security principal at a particular scope for the purpose of granting access.

WHO...WHAT...WHERE!!

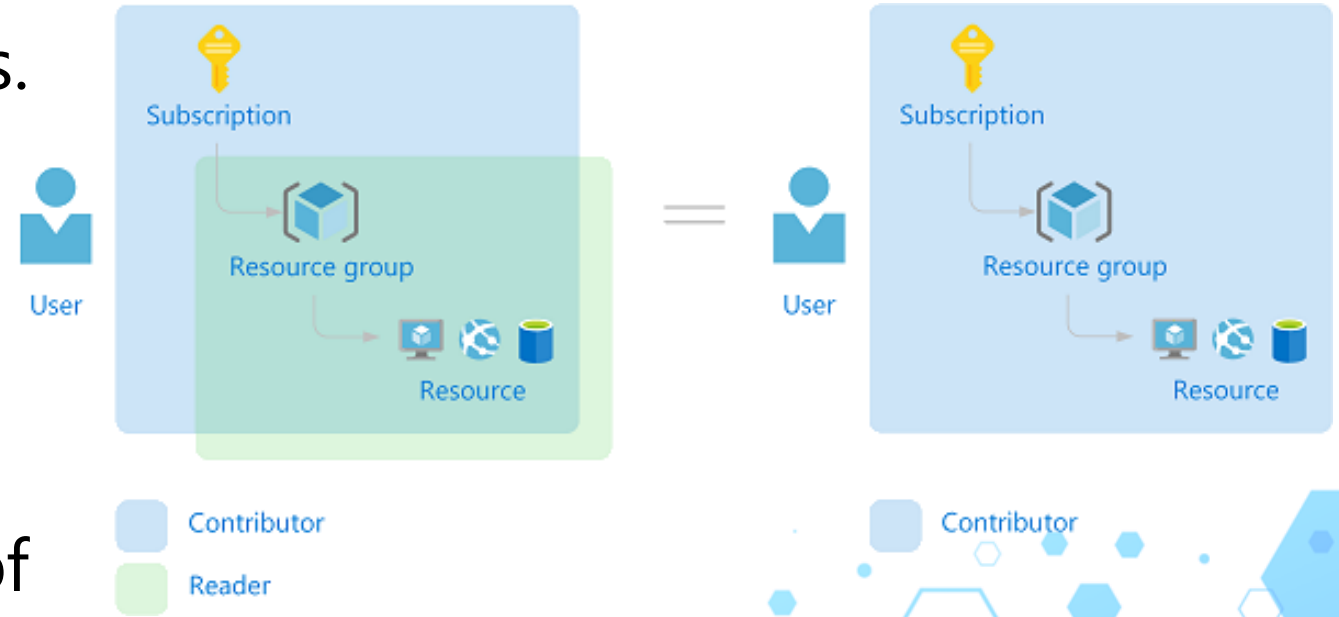


Multiple role assignments

Azure RBAC is an **additive model**, so your effective permissions are the sum of your role assignments.

Azure RBAC supports deny assignments.

A role assignment defines a set of actions that are *allowed*, while a deny assignment defines a set of actions that are *not allowed*.



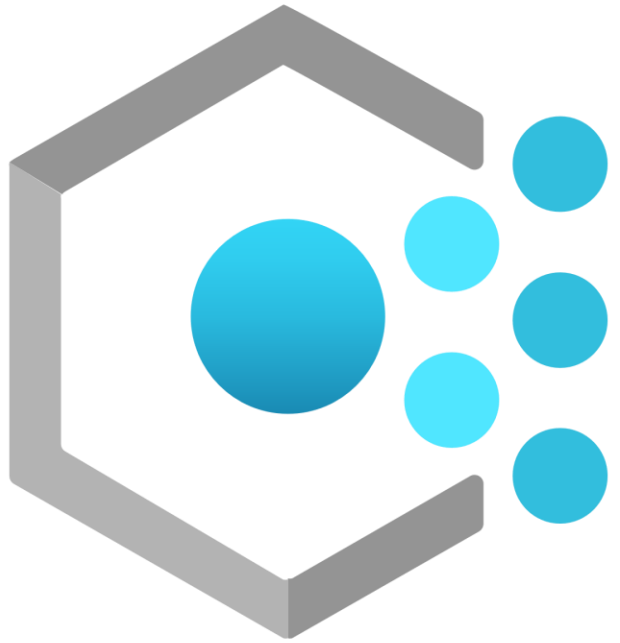
Azure
Meetup

DEMO

RBAC in Action!!!

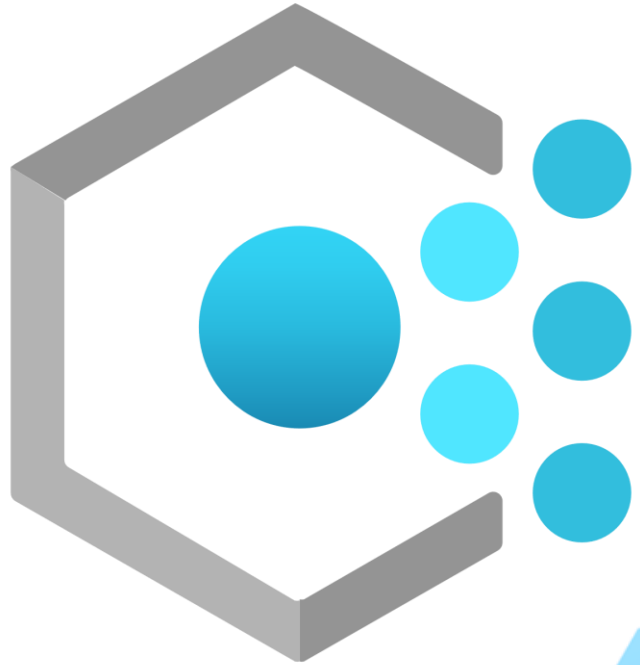
Azure
Meetup

Azure Policies



Azure
Meetup

Azure Policy helps to enforce organizational standards and to assess compliance at-scale.



What is Azure Policy

Azure Policy evaluates resources in Azure by comparing the properties of the resources to business rules.

The business rules, described in JSON format, are known as **Policy Definitions**.

Several business rules can be grouped together to form a **Policy Initiative**.

```json
{
    "properties": {
        "displayName": "Allowed locations",
        "description": "This policy enables you to restrict the locations yo
        "mode": "Indexed",
        "metadata": {
            "version": "1.0.0",
            "category": "Locations"
        },
        "parameters": {
            "allowedLocations": {
                "type": "array",
                "metadata": {
                    "description": "The list of locations that can be specif
                    "strongType": "location",
                    "displayName": "Allowed locations"
                },
                "defaultValue": [ "westus2" ]
            }
        },
        "policyRule": {
            "if": {
                "not": {
                    "field": "location",
                    "in": "[parameters('allowedLocations')]"
                }
            },
            "then": {
                "effect": "deny"
            }
        }
    }
}
```

Azure
Meetup

Evaluation Triggers

Resources are evaluated at specific times during the resource lifecycle, the policy assignment lifecycle, and for regular ongoing compliance evaluation.

A resource is created, updated, or deleted in a scope with a policy assignment.

A policy or initiative is newly assigned to a scope.

A policy or initiative already assigned to a scope is updated.

During the standard compliance evaluation cycle, which occurs once every 24 hours.

Policy definition

Policy metadatas (name, description, mode)

Policy Parameters

They values can change based on the assignment.

Policy Rule

Consists of **If** and **Then** blocks.

In the **If** block, you define one or more conditions that specify when the policy is enforced.

In the **Then** block, you define the effect that happens when the **If** conditions are fulfilled.

```
{
    "properties": {
        "displayName": "Allowed locations",
        "description": "This policy enables you to restrict the locations yo
        "mode": "Indexed",
        "metadata": {
            "version": "1.0.0",
            "category": "Locations"
        },
        "parameters": {
            "allowedLocations": {
                "type": "array",
                "metadata": {
                    "description": "The list of locations that can be specif
                    "strongType": "location",
                    "displayName": "Allowed locations"
                },
                "defaultValue": [ "westus2" ]
            }
        },
        "policyRule": {
            "if": {
                "not": {
                    "field": "location",
                    "in": "[parameters('allowedLocations')]"
                }
            },
            "then": {
                "effect": "deny"
            }
        }
    }
}
```

Rule sample

The rule **denies** any resource not of the **`Microsoft.Network/*`** type

```
"field": "type",
"notLike": "Microsoft.Network/*"
```

in any resource group whose name ends in "**`netrg`**".

```
"value": "[resourceGroup().name]",
"like": "*netrg"
```

```
{
    "if": {
        "allOf": [{
                "value": "[resourceGroup().name]",
                "like": "*netrg"
            },
            {
                "field": "type",
                "notLike": "Microsoft.Network/*"
            }
        ]
    },
    "then": {
        "effect": "deny"
    }
}
```

Azure Policy effects

Azure Meetup

- Append
- Audit
- AuditIfNotExists
- Deny
- DeployIfNotExists
- Disabled
- Modify

Policy Assignment

Policy assignments are used to define which resources are assigned which policies or initiatives.

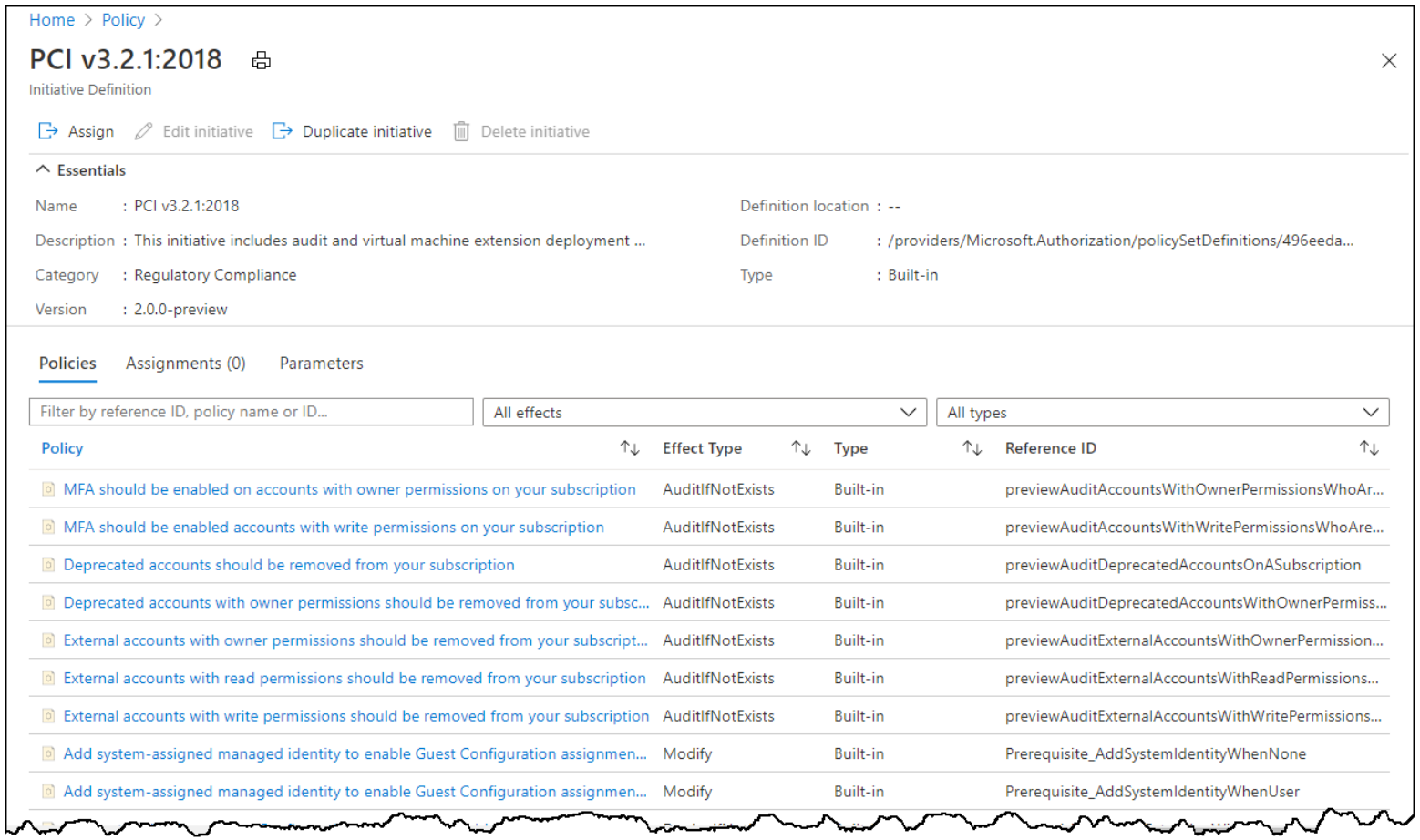
Policy Definition

Policy Parameters

```
{
    "properties": {
        "displayName": "Enforce resource naming rules",
        "description": "Force resource names to begin with DeptA and end with -LC",
        "metadata": {
            "assignedBy": "Cloud Center of Excellence"
        },
        "enforcementMode": "DoNotEnforce",
        "notScopes": [],
        "policyDefinitionId": "/subscriptions/{mySubscriptionID}/providers/Microsoft.Authoriza
        "parameters": {
            "prefix": {
                "value": "DeptA"
            },
            "suffix": {
                "value": "-LC"
            }
        }
    }
}
```

Policies and Initiatives

Azure Meetup

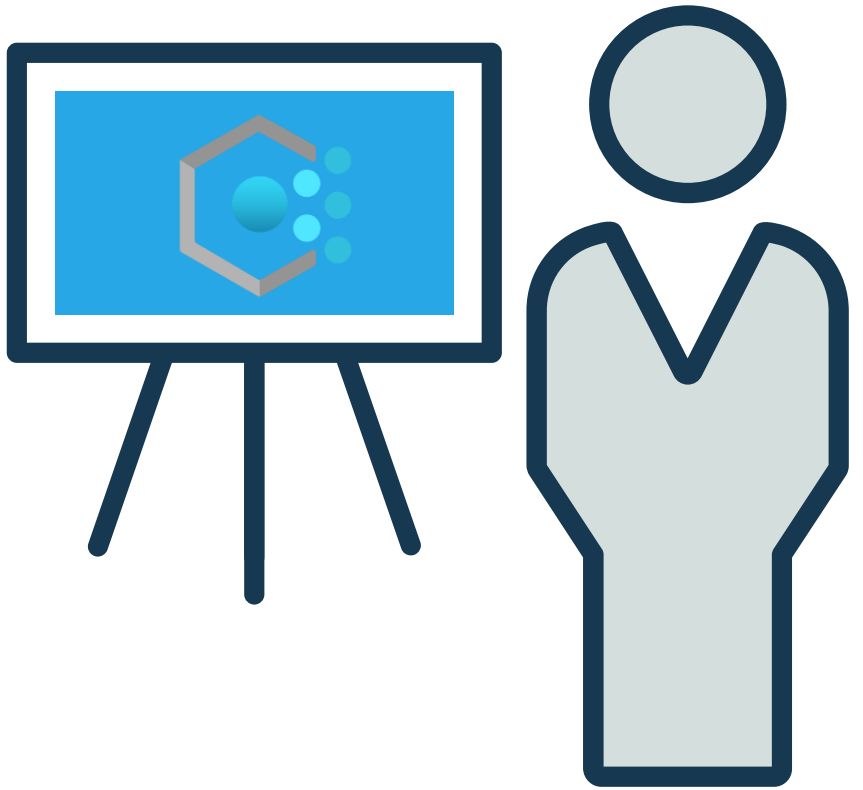


Initiatives enable you to group several related policy definitions to simplify assignments and management

Azure
Meetup

DEMO

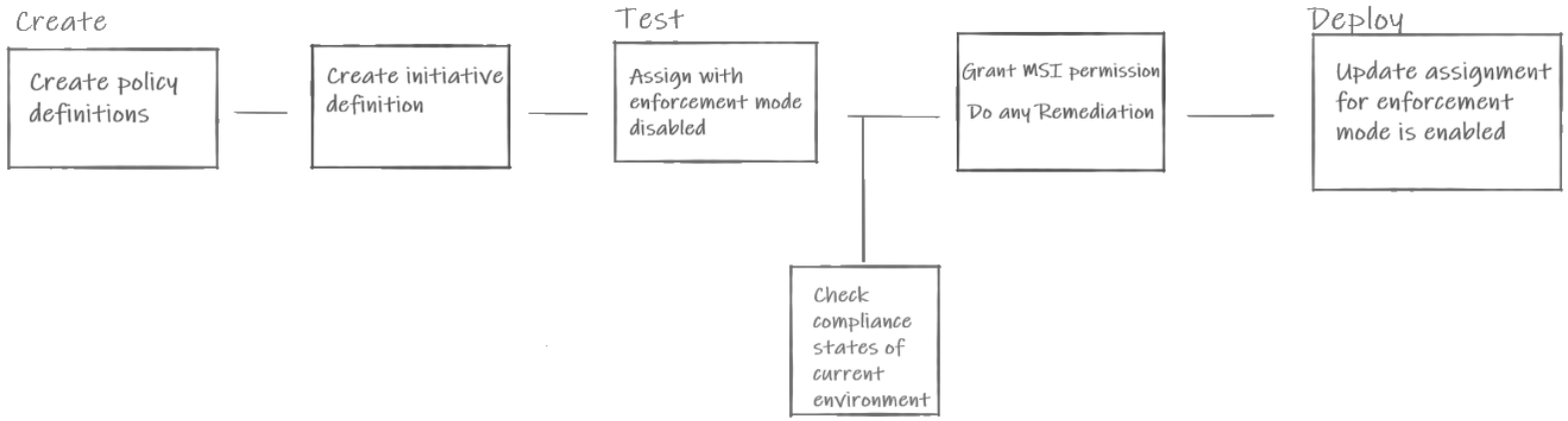
Policies overview!!

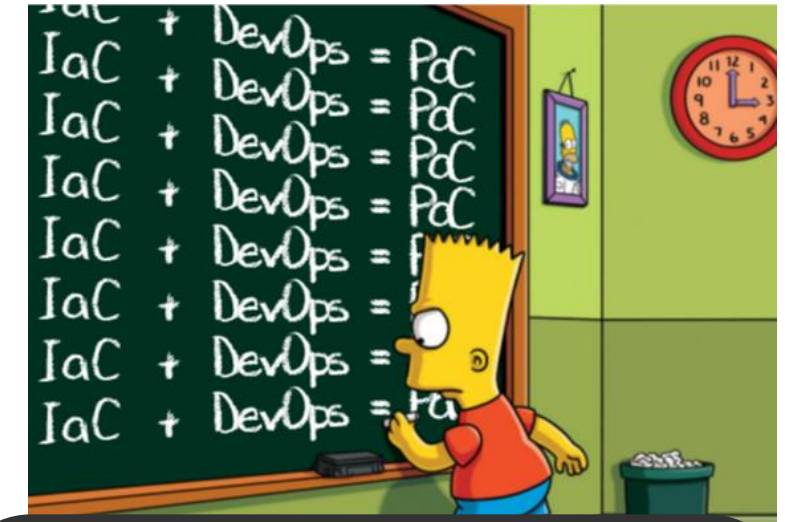


Azure Policy as Code

Azure Meetup

Keep your policy definitions in source control and whenever a change is made, test, and validate that change.





NEWS

Published date:
November 25, 2020

You can now export your Azure policies to GitHub directly from the portal!

Azure Policy vs RBAC

Azure Meetup

Azure Policy

- ✓ Evaluates state by examining properties on resources that are represented in Resource Manager and properties of some Resource Providers
- ✓ Doesn't restrict actions
- ✓ Ensures that resource state is compliant to your business rules without concern for who made the change or who has permission to make a change

RBAC

- ✓ Focuses on managing user actions at different scopes
- ✓ Even if an individual has access to perform an action, if the result is a non-compliant resource, Azure Policy still blocks the create or update

Azure Meetup

Connect with me on LinkedIn

linkedin.com/in/massimobonanni

Thanks for your attention!!!!!

Massimo Bonanni

Azure Technical Trainer

massimo.bonanni@microsoft.com
@massimobonanni

Microsoft Certified: Azure Fundamentals
Microsoft Certified: Azure Data Fundamentals
Microsoft Certified: Azure AI Fundamentals
Microsoft Certified: Azure Developer Associate
Microsoft Certified: Azure Administrator Associate
Microsoft Certified: Azure AI Engineer Associate
Microsoft Certified: Azure Solutions Architect Expert
Microsoft Certified: Azure DevOps Engineer Expert
Microsoft Certified: Azure IoT Developer Specialty
Microsoft Certified Trainer MCT 2020 - 2021

References

Azure Meetup

- Role Base Access Control documentation
  https://docs.microsoft.com/en-us/azure/role-based-access-control/overview
- Role Base Access Control - Learning Paths
  https://docs.microsoft.com/en-us/learn/browse/?expanded=azure&products=azure-rbac
- Azure Policy documentation
  https://docs.microsoft.com/en-us/azure/governance/policy/overview
- Azure Policy – Learning Paths
  https://docs.microsoft.com/en-us/learn/browse/?expanded=azure&products=azure-policy
- Design Azure Policy as Code workflows
  https://docs.microsoft.com/en-us/azure/governance/policy/concepts/policy-as-code
- Tutorial: Implement Azure Policy as Code with GitHub
  https://docs.microsoft.com/en-us/azure/governance/policy/tutorials/policy-as-code-github
- Azure Policy – GitHub
  https://github.com/Azure/azure-policy

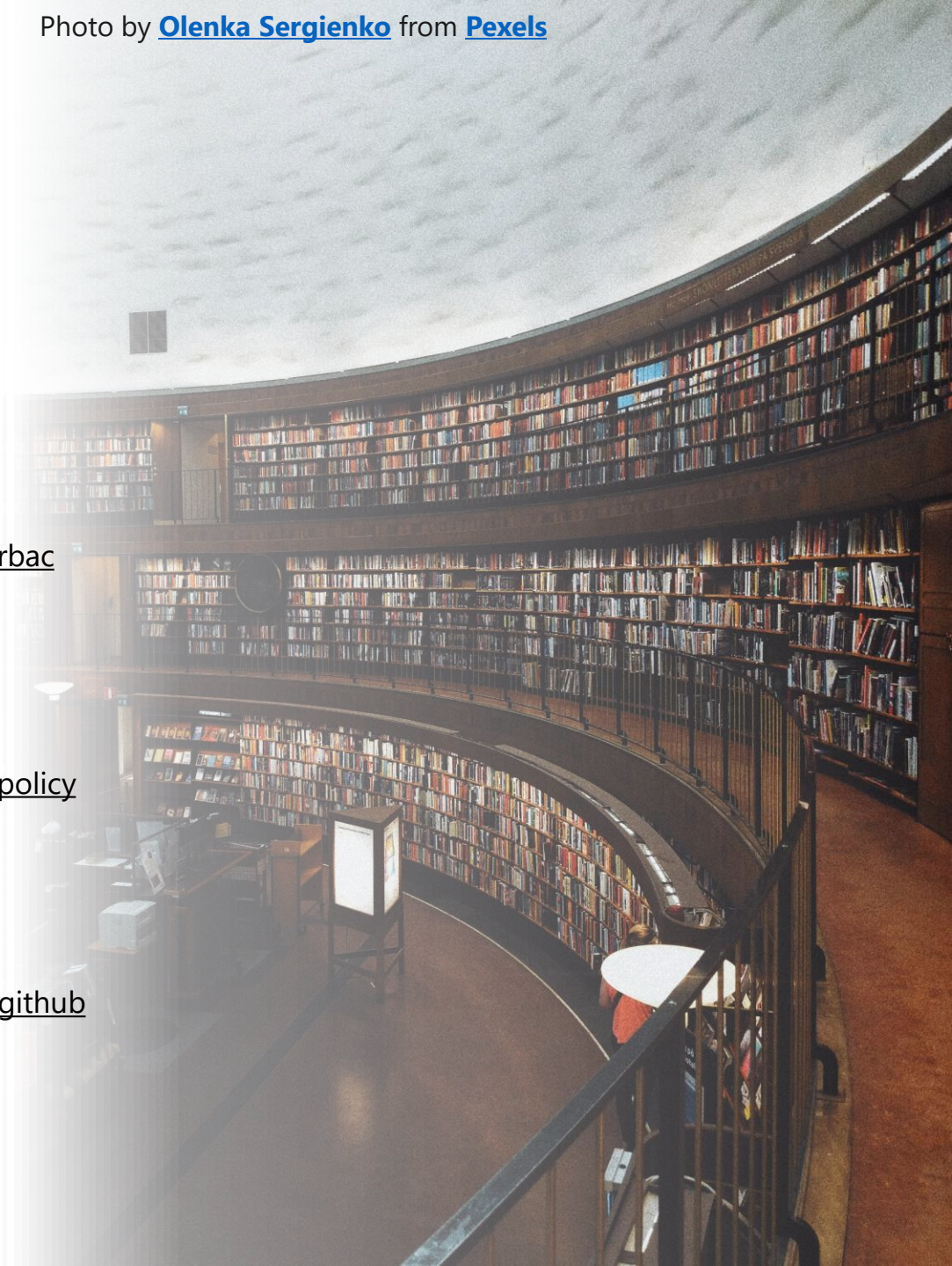


Photo by Olenka Sergienko from Pexels