A STRAIGHTFORWARD GUIDE

Azure Governance

Ricardo M. Martins

Table of Contents

Table of Contents

INTRODUCTION

Hello, I am Ricardo Martins

I am married to Patricia, father of Anna Clara, Maria Luísa, and Pedro Henrique. I am someone passionate about building technical education that helps people advance in their careers, and over the years, I have been recognized for this.

Since the early 2000s, I have been working with Linux, Servers, and Networks, starting my career as an intern at a small internet service provider in my hometown.

Later on, I became a Systems Engineer working with IT Infrastructure in various companies across different sectors, from startups to large corporations.

In 2012, I began dealing with cloud-based architectures, where I have been automating, designing, and developing cloud systems, as well as leading efforts in reliability, monitoring, alerts, automated deployment, fault resilience, and capacity management.

In 2015, I started working at Microsoft, where I developed strong skills in different roles, such as Cloud Solutions Architect, Technical Instructor, and Systems Engineer.

In early 2023, I joined Red Hat to work as a Black Belt in native cloud technologies.

Over the past 20 years, I have been educating, designing, training, and building cloud solutions for a diverse and challenging range of clients.

- Ricardo Martins

CHAPTER 01

Governance Overview

Why is governance needed?

Companies are adopting the cloud to be more agile and save money. There is pressure to transform and innovate digitally so that you no longer have time to focus on your own infrastructure. You want to focus on making your customers happy by providing high-quality services with the support of your engineering teams. So there is a natural shift to DevOps in a cloud environment, where engineers will more quickly provide the resources needed to support a solution.

However, this agility and easy access to resources come at a price and many companies are struggling to control this Cloud Sprawl. We have seen this before, in early 2000, with the introduction of virtualization and the proliferation of virtualization.

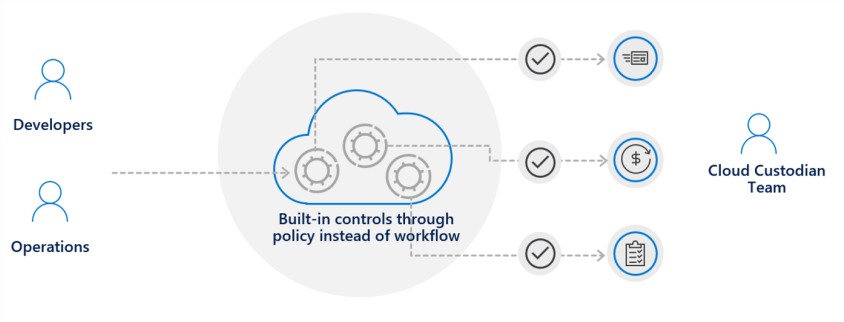
How did the industry react to this expansion? We jumped in front of the developers and the operations teams and stopped them before things got out of hand. We then introduced a formal process for these teams to follow where they should fill out a form so that the infra team could set up everything and in 2 weeks they would have access to their environment.

CHAPTER 01 CONTINUED

However, this approach in the cloud age slows things down and you sacrifice speed to be in control.

In a cloud-native governance model, you get both speed and control at the same time. So instead of jumping in front of the DevOps team to make sure they're doing the right things, the cloud platform itself will enforce that control on your behalf. This allows them to have full access to the platform through a self-service model that is essential to maintain agility and speed.

You can guarantee that your teams will deploy only approved resources and anything outside these rules will be effectively denied. That way, you keep your costs predictable and more in line with your budget.



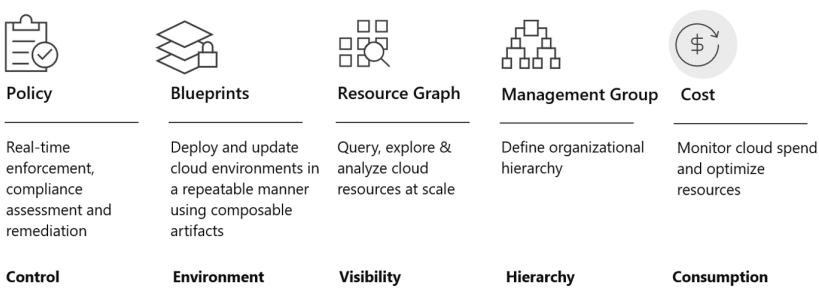
Aligned with governance it's important to have a well-defined structure around responsibilities across different teams, especially if you are migrating from a traditional approach to a cloud approach. That said, you should take a look into those references to help you mature team structures and align responsibilities within them:

- Mature team structures
- Align responsibilities across teams

CHAPTER 01 CONTINUED

Native Features for Cloud Governance

The resources we have in the Azure portfolio to deliver this level of control are what you see in the image above. All of these are native features of the platform, in other words, nothing here is a different product that you need to purchase or deploy in your cloud environment. The moment you create your first subscription, these features are there for you to use right away and are completely free.



Policy

This is where you will define what can and cannot be deployed in the cloud. He will constantly check your signatures and resources to ensure that everything complies with corporate rules.

Blueprints

It helps you to configure your cloud environment so that it is managed properly and deployed in a repeatable manner. It allows a kind of implementation of governance as a code. Subscriptions are made available to development teams or departments as they are created. The goal here is that, when teams are presented with a subscription controlled by Blueprints, the amount of time they need to take from initial subscription settings, permits, policies, etc. to the implementation of the project in production decreases dramatically.

Resource Graph

This is a Big Data technology where we bring the configurations of all your resources from your cloud environment and offer you, through a structured query language, the ability to explore your environment very quickly, allowing visibility at scale over all your environment.

Management Group

If your organization has many subscriptions, you may need a way to efficiently manage access, policies and compliance for those subscriptions. This can be done through the Management Group.

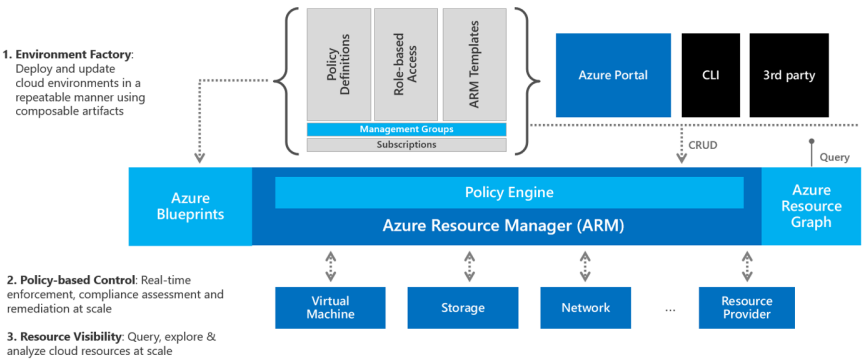
Cost Management

Helps you understand your Azure invoice, monitor and control spending, and optimize resource usage. It allows you to analyze costs, create and manage budgets, export data, examine recommendations and act on them.

CHAPTER 01 CONTINUED

Governance Architecture in Azure

Let's add these governance features to a diagram:



Think of multiple development teams or business units that want to consume Azure resources to meet their specific business requirements. How many of these people, especially DevOps, will readily know how to properly design and configure network components?

Ultimately, you want a quick and efficient way to deploy and update cloud environments in a repeatable way using combinable artifacts.

You would start by setting up your blueprints. And these blueprints would have all the basic and necessary components for an entire solution, such as their policy definitions, RBAC functions, ARM models, number of signatures needed and the preliminary groups already granted the appropriate access levels to the resources in that blueprint, such as their VMs, storage accounts, network components and any other resource providers.

Finally, you can take advantage of the Resource Graph to check the environment and ensure, for example, that the number of resources is in accordance with your budget.

CHAPTER 01 CONTINUED

If the cloud is something new for you, I have some suggestions to help you start to build your technical skills:

- Building Technical Skills
- Azure Fundamentals Part 1: Describe Cloud Concepts
- Azure Fundamentals Part 2: Describe Azure Architecture and Services
- Azure Fundamentals Part 3: Describe Azure Management and Governance
- Prerequisites for Azure Administrators
- Build great solutions with the Microsoft Well-Architected Framework

CHAPTER 02

Azure Active Directory (Entra ID)

What is the Azure Active Directory? (Now called Microsoft Entra ID)

In order to begin to understand in more detail the main services related to Azure Governance, it is important to start by talking about the relationship between Azure Active Directory (AAD) - Microsoft Entra ID and subscriptions. It will not be covered all the details about it here since this is not the purpose of this document. However, here we will see the basics of how it works and the difference between Azure Active Directory (AAD) and Active Directory (AD).

AAD (Microsoft Entra ID) is the cloud-based identity and access management service that will allow you to grant access to users, groups, and applications on Azure services as well as allowing you to define how they will use Azure resources through the functions that you will assign to them. In this way, it will take on the role of managing authorization and authentication for Azure services.

When creating an Azure subscription, an AAD tenant is automatically created. The tenant is nothing more than the representation of your company's domain within Azure Active Directory. Note that by default you will always get a name.onmicrosoft.com that you can then customize for yourdomain.com. Within your AAD tenant, you will have your AAD directory which is where you will create your users and groups. Note that you can also sync your existing users in your existing Active Directory to Azure Active Directory via Azure AD Connect, but this topic will not be covered here.

So basically, the main difference from Active Directory to Azure Active Directory is that AAD aims to work exclusively on the authorization and authentication of its users, groups, and applications in Azure services. In contrast, Active Directory performs authorization and authentication in the on-premises environment in addition to many other tasks such as managing GPOS and Windows servers. In this link, there is a broader comparison between them.

- Azure AD Identity Governance
- Azure AD Access Reviews

CHAPTER 03

Naming Standards

A naming strategy for use with Azure resources is important within the governance approach. The use of some naming conventions will assist in the location and management of resources as well as in the association of costs with the business areas. There is very comprehensive documentation on defining naming conventions available here:

- Define your naming convention

Likewise, you can find a list of recommended abbreviations for the most diverse types of Azure resources at this link:

- Recommended abbreviations for Azure Resource Types

CHAPTER 04

Azure Subscription

As stated earlier, when creating an Azure subscription an AAD tenant is automatically created for you. With this, after creating and/or synchronizing users in Azure Active Directory, you can now allow your ADF users to subscribe to your subscription and its existing resources.

According with the size of your cloud environment, you can also create additional subscriptions or associate other existing subscriptions with your Azure Active Directory tenant. Having at least two subscriptions, one for the production environment and the other for non-production ones, is a good practice for segregation of the environment and for scalability.

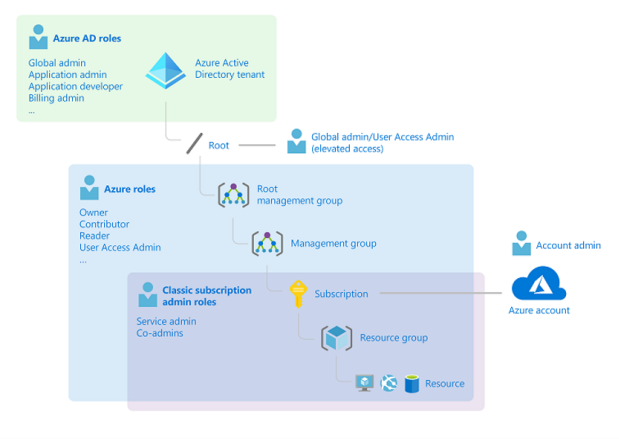
An important point to be considered is about permissioning is that there are two types of functions/attributions that are distinct but totally related to each other:

- Azure Roles: Azure Roles use Role Based Access Control (RBAC) and are granted in the context of Azure resources within a subscription. There are three basic roles of Owner, Collaborator and Reader. In addition to them there are more than 70 other roles that are more related to services specifically, here you can see the list with all. In addition to the native functions, you may want to create your own custom roles and maximize the type of control you want to apply.

- Azure Active Directory roles: Azure Active Directory roles are used exclusively for the management of Azure Active Directory resources.

CHAPTER 04 CONTINUED

This image can help you understand a little about how the functions of Azure and Azure Active Directory are related:



An Azure subscription has a trust relationship with the AAD to authenticate and authorize users, services and devices.

It is important to know that the same AAD tenant can have multiple subscriptions trusting him, but each signature can only confirm on a single AAD tenant. It means that you can have the same user base on the AAD tenant for different subscriptions.

A subscription is a logical container for your resources and each resource is associated with only one signature. They are directly related to billing and payment.

The data in the subscriptions remains for a while after being canceled, and the subscriptions themselves are usually visible, even after being canceled in the Portal and in the APIs. There is information about the cancellation process in the documentation available here.

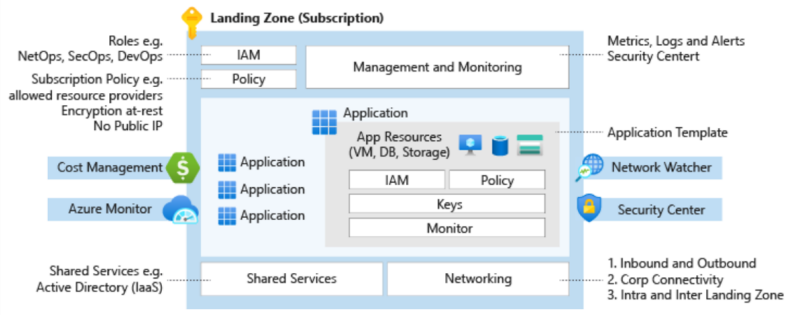
A subscription serves different purposes because it is a legal contract, a payment contract, a scaling limit and, an administrative limit. All details are described in this link.

It is important to define an architecture for the use of subscriptions so that there is a better organization and management of resources, especially in the segregation of permission and control of existing limits on subscriptions. To help with this, there is a documented decision guide that is super interesting in understanding the best way to model your organization and define subscription design strategies.

CHAPTER 04 CONTINUED

Landing Zones Overview

As described in this link, the recommendation is that there are at least two signatures, one for the production environment and the other for the non-production environment. Depending on the size of your environment or the strategy of your company, it may be necessary to create more signatures and in addition to combine the design of signatures with the definition of the landing zone to be created.

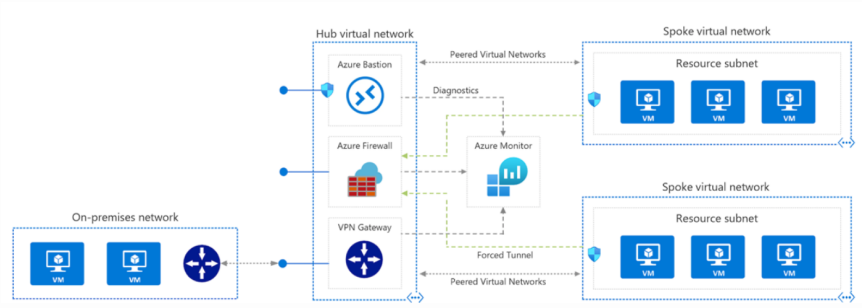


The Microsoft Cloud Adoption Framework describes in detail several topics over the enterprise-scale landing zone architecture, which offers a modular design and not only makes it simple to deploy existing and new applications but also allows organizations to start with a lighter deployment implementation and scale depending on their business needs.
Basically, the landing zone will deal with a set of considerations and recommendations based on some design areas:

- Enterprise Agreement (EA) enrolment and Azure Active Directory tenants
- Identity and access management
- Management group and subscription organization
- Network topology and connectivity
- Management and monitoring
- Business continuity and disaster recovery
- Security, governance, and compliance
- Platform automation and DevOps

CHAPTER 04 CONTINUED

The choice of network topology to be used is important for the process of governance definition. For example, the Hub and Spoke topology may be inserted in the context of subscriptions as follows:

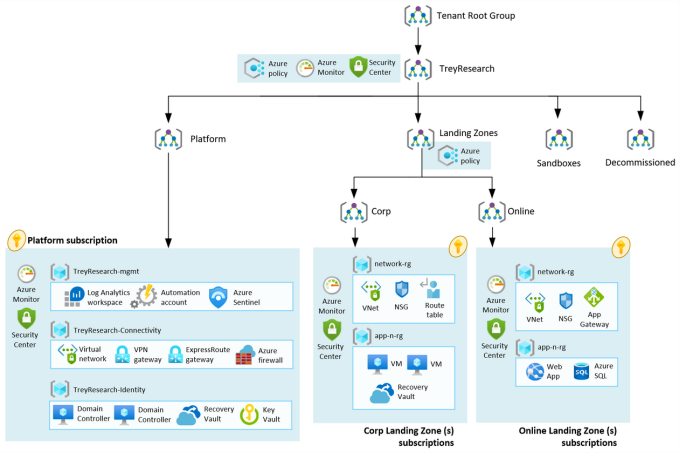


- The first subscription for **shared services** (Hub Virtual Network)
- A second subscription for the **production** environment (Spoke Virtual Network - at the top right)
- A third subscription for the **non-production** environment (Spoke Virtual Network - at the bottom right)

Some references about Hub and Spoke topology:

- Hub Spoke Topology on Azure
- Define an Azure network topology

Approaching the Enterprise Scale Landing Zone, the architecture above could be translated into the architecture below to bring the "enterprise-scale" ability to the environment:



As you can note, this architecture adopts the usage of different Management Groups and Subscriptions to split the environment into two main groups: Platform and Landing Zones, this principle suggests production environments transitioned to business units and workload units. This allows workload owners to have more control and autonomy of their workloads within the guardrails established by the platform foundation.

Currently, enterprise-scale offers different reference implementations, which all can be scaled without refactoring when requirements change over time.

- Enterprise-Scale Reference Implementation

CHAPTER 05

Resource Groups

Resource groups are the approach that allows you to group a collection of objects into logical groups, facilitating provisioning, monitoring, access, and cost control. The benefit of using resource groups is precisely the grouping of resources that are part of the same application or share the same life cycle from creation to de-provisioning.

The underlying technology that empowers resource groups is Azure Resource Manager (ARM). Just for information, ARM was created to replace the previous technology called Azure Service Manager (ASM) that powered the old Azure management portal. In ASM, users created resources in an unstructured way, leading to many challenges in tracking those resources or understanding their dependencies. You can see more details here:

- Resource Manager and Classic Deploy

Since ARM became available, these and other challenges have been addressed in addition to providing a new set of application programming interfaces (APIs) for provisioning resources in Azure. ARM requires resources to be placed in groups of resources, allowing the logical grouping of related resources.

- What is a Resource Group
- Principles of Resource Groups

CHAPTER 06

Resource Tags

Associated with a good name pattern strategy is the use of tags on objects. Tags are applied to resources, resource groups, and signatures to logically organize them into a taxonomy. The tags consist of a name/value pair. Here are some examples of tags that you can consider using in your environment:

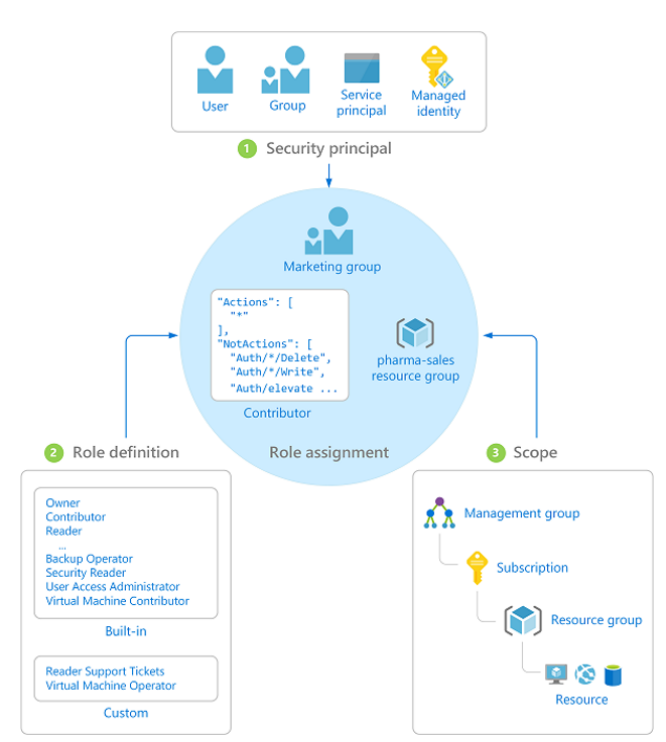
Name ⓘ		Value ⓘ	Resource
AppName	:	SpecialOrders	Virtual machine
CostCenter	:	0224 - Infrastructure R&D	Virtual machine
Owner	:	tim@tailwindtraders.com	Virtual machine
Environment	:	Test	Virtual machine
Impact	:	High-impact	Virtual machine

Going beyond support in governance, the use of tags can also be used in automation and cost control strategies. Use the links below to access a tag decision and resource naming guide and excellent material to assist you in defining the tagging strategy:

- Define your tagging strategy
- Resource Naming and Tagging Decision Guide

CHAPTER 07

Role Based Access Control



Role Based Access Control (RBAC) allows you to separate tasks within your team and grant only the amount of access that users need to perform their tasks. Instead of giving everyone unrestricted permissions on your subscription or Azure features, you can only allow certain actions within a given scope.

CHAPTER 07 CONTINUED

Security Principal

A security principal is an object that represents a user, group, service principal, or managed identity that is requesting access to Azure resources.

- Service Principal: a security identity used by applications or services to access specific Azure features. You can think of it as a user identity (username and password or certificate) for an application.

- Managed Identity: an identity in Azure Active Directory that is automatically managed by Azure. Typically, you use managed identities when developing cloud applications to manage credentials for authenticating to Azure services.

Role Definition

A role definition is a collection of permissions. Sometimes, it is just called role. A role definition lists operations that can be performed, such as reading, writing, and deleting. The roles can be high-level, as an owner, or specific, as a virtual machine reader. You can create custom roles if none of the existing integrated roles does not meet your organization's specific needs.

Scope

The scope is the limit to which access applies. When assigning a role, you can further limit the actions allowed by defining a scope. This is useful if you want to make someone a Website Contributor for example, but only for a group of resources.

Scopes can be Management Groups**, Subscriptions, Resource Groups, or a Resource itself. It is important to note that RBAC permissions have a top-down hierarchy, which means that when defining a role definition in a higher scope, it will be replicated to objects in the lower scope. So, if you grant Contributor permission to someone at the subscription level, that permission will be inherited in all resource groups under this subscription, as well as resources.

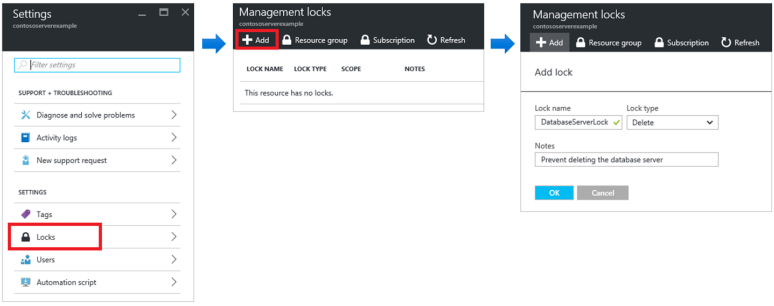
- What is Azure Role-Based Access Control?
- Manage access to an Azure subscription by using Azure role-based access control

*** Management Group is a feature of Azure to facilitate the management of access and policies in environments with multiple subscriptions. Will be covered in more detail in the chapter 13.*

CHAPTER 08

Resource Locks

Resource Locks are a feature that allows you to avoid accidentally deleting or modifying critical resources. The lock overrides any permissions the user might have.



When you apply a lock on a parent scope, all resources within that scope inherit the same lock. Even the features you add later inherit the parent's lock. The most restrictive block on inheritance takes precedence. Unlike role-based access control, you use management locks to apply a restriction to all users and roles

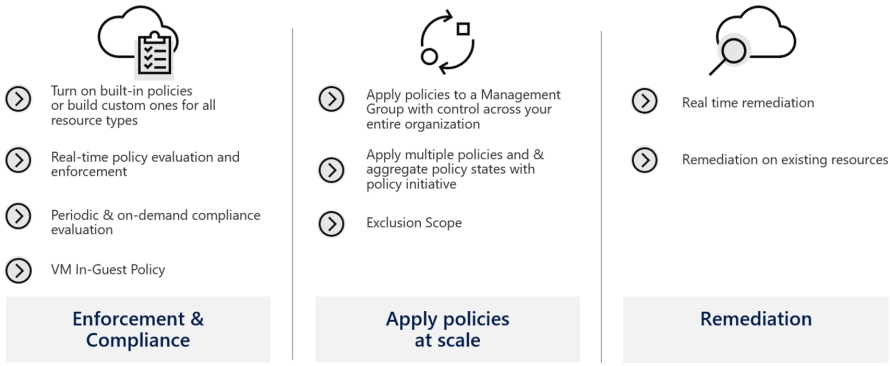
Resource Manager locks apply only to operations that take place in the management plan, which consists of operations sent to https://management.azure.com. Locks do not restrict how resources perform their own functions. Resource changes are restricted, but resource operations are not restricted. For example, a ReadOnly lock on an SQL database prevents you from deleting or modifying the database, but it does not prevent you from creating, updating, or deleting data in the database. Data transactions are allowed because these transactions are not sent to https://management.azure.com

Applying ReadOnly can lead to unexpected results because some operations that look like reading operations actually require additional actions. For example, placing a ReadOnly lock on a storage account prevents all users from listing keys. The operation of list keys is handled through a POST request because the returned keys are available for write operations. For another example, placing a ReadOnly lock on an application service resource prevents Visual Studio Server Explorer from displaying files for the resource because that interaction requires to write access.

- Protect your Azure resources with a lock

CHAPTER 09

Azure Policy



Policy is the backbone of Azure implementation and compliance. Compliance is an evolving scenario that you will always need to assess and adjust according to the needs of the moment. You also need to carefully plan your policies so as not to interrupt other units that require a more flexible policy than the one you have defined.

Through policies, you can control the types of resources that can be provisioned. Or, you can restrict the locations where resources can be provisioned. Unlike RBAC, the policy is a standard system of explicit permission and denial.

Azure Policy is a service that you use to create, assign and manage policy definitions. Policy definitions impose different rules and actions on your resources, so that those resources remain in compliance with your corporate standards and service level agreements.

The policy focuses on the properties of resources during deployment and for existing resources. It performs an assessment of your resources, checking those that do not conform to the policy definitions you have. A very interesting new feature is the VM guest policy which extends the policy's capacity to the resources running within your VMs.

CHAPTER 09 CONTINUED

Azure Policy best practices

Ask yourself these 3 questions and work from them when defining your policies:

- **What drives your need for policy?**
  - Regulatory Compliance
  - Controlling cost
  - Standards & Tagging
  - Maintain security and performance consistency
  - Enforce enterprise-wide design principles

- **Who owns the policy settings?**
  - “Initiative" owners
  - Security Architect
  - Cloud Architect
  - Cloud Engineers

- **What is involved in defining a new policy or refining an existing one?**
  - Research or gather evidence on the impact of a particular configuration on a particular fundamental (like cost or security)
  - What-if analysis of enforcing configuration in a particular manner
  - Assess the current state of compliance to understand the impact of the new policy and what exceptions are needed
  - Roll out a new policy in phases
  - Understand the applications & teams who are non-compliant
  - Rollout remediation in stages via SafeDeploy practices

These questions need to be asked from time to time as compliance is an evolving thing. You need to adjust your policies according to your current priorities, not only for compliance but also for different projects that might require more powerful resources deployed that are currently blocked by policy, for example.

CHAPTER 09 CONTINUED

Governance suggested policies

Here is a list of suggested policies you can apply in your environment in order to help in your governance approach.

- Compute
  - **Allowed virtual machine size SKUs:** This policy enables you to specify a set of virtual machine size SKUs that your organization can deploy.
    - Click here to see on Azure Portal
    - Click here to see the JSON file

- General
  - **Allowed locations:** This policy enables you to restrict the locations your organization can specify when deploying resources. Use to enforce your geo-compliance requirements. Excludes resource groups, Microsoft.AzureActiveDirectory/b2cDirectories, and resources that use the 'global' region.
    - Click here to see on Azure Portal
    - Click here to see the JSON file

  - **Allowed locations for resource groups:** This policy enables you to restrict the locations your organization can create resource groups in. Use to enforce your geo-compliance requirements.
    - Click here to see on Azure Portal
    - Click here to see the JSON file

  - **Allowed resource types:** This policy enables you to specify the resource types that your organization can deploy. Only resource types that support 'tags' and 'location' will be affected by this policy. To restrict all resources please duplicate this policy and change the 'mode' to 'All'.
    - Click here to see on Azure Portal
    - Click here to see the JSON file

CHAPTER 09 CONTINUED

- **Audit resource location matches resource group location:** Audit that the resource location matches its resource group location
  - Click here to see on Azure Portal
  - Click here to see the JSON file

- **Audit usage of custom RBAC rules:** Audit built-in roles such as 'Owner, Contributor, Reader' instead of custom RBAC roles, which are error-prone. Using custom roles is treated as an exception and requires a rigorous review and threat modeling
  - Click here to see on Azure Portal
  - Click here to see the JSON file

- **Custom subscription owner roles should not exist:** This policy ensures that no custom subscription owner roles exist.
  - Click here to see on Azure Portal
  - Click here to see the JSON file

- **Not allowed resource types:** Restrict which resource types can be deployed in your environment. Limiting resource types can reduce the complexity and attack surface of your environment while also helping to manage costs. Compliance results are only shown for non-compliant resources.
  - Click here to see on Azure Portal
  - Click here to see the JSON file

- Security

Please note that if you decide to enable the Azure Security Center built-in initiatives, be on the lookout for overlapping conflicts. See here the Azure Policy built-in definitions for Azure Defender for Cloud

- **A maximum of 3 owners should be designated for your subscription:** It is recommended to designate up to 3 subscription owners in order to reduce the potential for breach by a compromised owner.
  - Click here to see on Azure Portal
  - Click here to see the JSON file

- **MFA should be enabled on accounts with owner permissions on your subscription:** Multi-Factor Authentication (MFA) should be enabled for all subscription accounts with owner permissions to prevent a breach of accounts or resources.
  - Click here to see on Azure Portal
  - Click here to see the JSON file

- **Subscriptions should have a contact email address for security issues:** To ensure the relevant people in your organization are notified when there is a potential security breach in one of your subscriptions, set a security contact to receive email notifications from the Security Center.
  - Click here to see on Azure Portal
  - Click here to see the JSON file

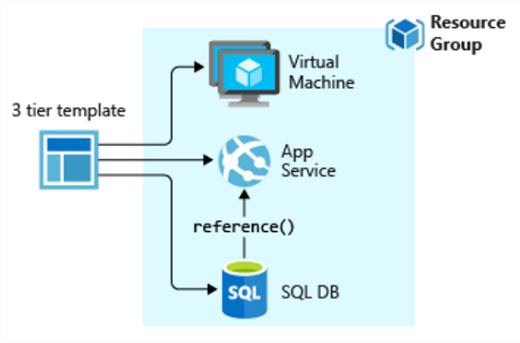
- **There should be more than one owner assigned to your subscription:** It is recommended to designate more than one subscription owner in order to have administrator access redundancy.
  - Click here to see on Azure Portal
  - Click here to see the JSON file

- Tags
  - **Require a tag on resource groups: Enforce the existence of a tag on resource groups.**
    - Click here to see on Azure Portal
    - Click here to see the JSON file

  - **Inherit a tag from the resource group if missing:** Adds the specified tag with its value from the parent resource group when any resource missing this tag is created or updated. Existing resources can be remediated by triggering a remediation task. If the tag exists with a different value it will not be changed.
    - Click here to see on Azure Portal
    - Click here to see the JSON file

CHAPTER 10

ARM Templates



Azure Resource Manager Templates are JSON files used to automate the deployment of Azure environments using infrastructure as code. With the use of infrastructure as code, the code repository for your project's applications now also has the way to deploy all the infrastructure required by your application in a coded, repeatable and versioned manner in the same way as the applications themselves. Through ARM Templates, you automate the entire deployment of your environment, from the creation of the network, storage, virtual machines, and installation of dependencies to the deployment of the application itself in an orchestrated way.

If you are interested in knowing more about ARM Templates and how to use them, access the documentation available at https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview. We currently have about 1000 templates ready and available for use in Azure Quickstart Templates, check out https://azure.microsoft.com/en-us/resources/templates/

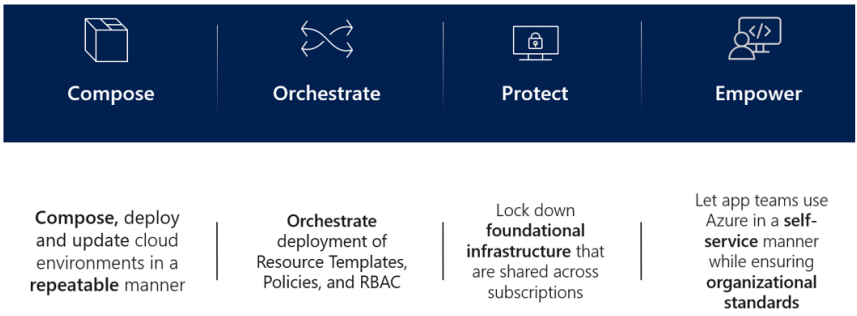
Recently was introduced a new language for developing ARM templates. The language is named Bicep, and is currently in preview. Bicep and JSON templates offer the same capabilities. You can convert templates between the two languages. Bicep provides a syntax that is easier to use for creating templates. For more information, see What is Bicep (Preview)

In another hand, if you are more familiar with opensource tools like Terraform to automate deployments on your IaC strategy, you can find a lot of useful resources here.

- Deploy and manage resources in Azure by using JSON ARM Templates

CHAPTER 11

Azure Blueprints



How can you bring everything that has already been discussed in a structured way, so that you can configure your environment in a consistent and automated way, at scale in the shortest possible time?

For a developer, there is a lot to do when setting up an Azure subscription for the first time. As you might say, Azure is an empty canvas and if you are not an artist, there is a lot to draw. And trying to fill that screen with a base image is what Blueprint really is about. He tries to create a fundamental foundation for the appearance of his environment.

There is no reason for your DevOps team to become a ninja on the Azure network, they just need to focus on their code, business logic, etc. and that's it.
What happens today is that you provide them with a giant document with all the required specifications and organize several meetings with them to make sure that they understand and follow those specifications, or just do everything for them, which increases the time of implementation, as there would be an alignment of other DevOps Engineer waiting for your environment to be provisioned.

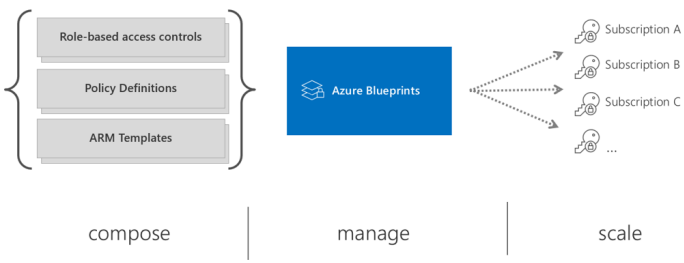
CHAPTER 11 CONTINUED

So, here are some of the main challenges for customers when designing and configuring the governance of their subscriptions:

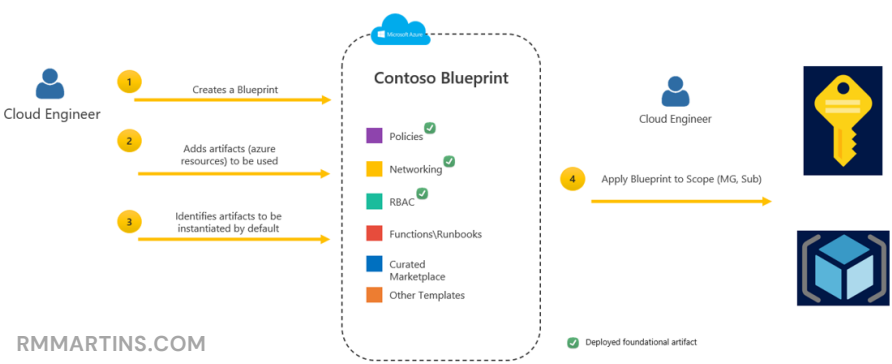
- **Challenging to configure the basic infrastructure:** It becomes complex to create and redistribute the infrastructure.
- **Inability to create governed signatures:** The absence of a centralized way of defining and ensuring that what is created or made available in a signature will be applied. The client uses a ton of scripts to try to do this.
- **Protection of critical resources:** Subscription owners can modify resources and remove policies in violation of best practices defined by cloud architects. Therefore, to address these key challenges faced by customers, Azure Blueprints was created where you have an automated and easy-to-deploy solution to help set up Azure Subscriptions in line with a governance strategy.

Therefore, to address these key challenges faced by customers, Azure Blueprints was created, providing an automated and easy-to-deploy solution to assist in configuring Azure Subscriptions in line with a governance strategy.

Azure Blueprints allows you to implement Governance as Code.

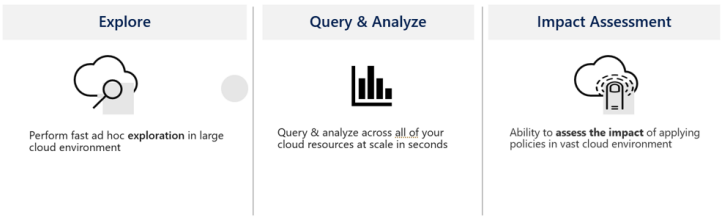


How Azure Blueprints works:



CHAPTER 12

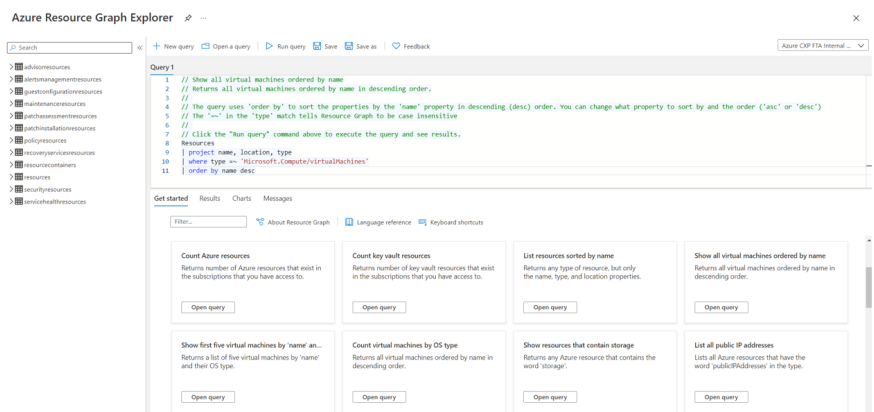
Azure Resource Graph



There are a few challenges today when it comes to getting visibility of all your resources properties in all subscriptions and understanding how they can be impacting negatively your organization, being from a cost or a security perspective.

- Inability to view resources and their properties across subscriptions
- Query resources (without worrying about timeouts & throttling), including filtering, grouping, and, sorting of resources by resource properties

The Resource Graph provides a way to interactively explore resources, so you can assess the impact of applying policies in a vast cloud environment. Resource Graph is the tool that comes to cover those needs allowing you to query and explore your resources in real-time.



- What is the Resource Graph?

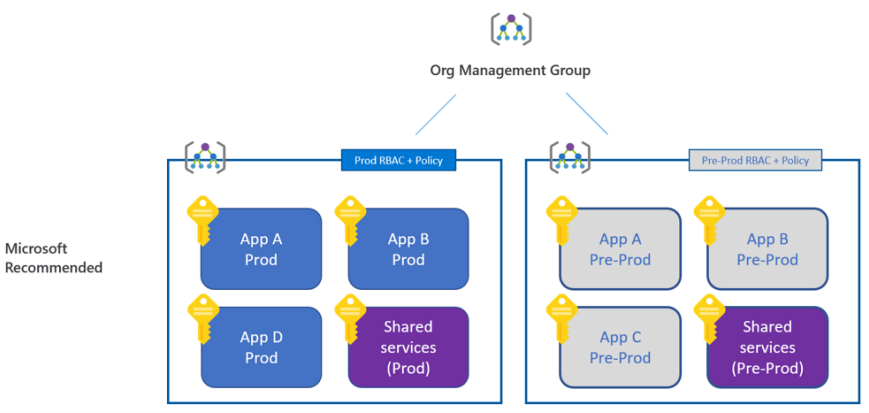
CHAPTER 13

Management Groups

In terms of the best practices of setting up governance in your environment structuring your hierarchy and organizing your resources is the first critical step.

In terms of the best practices of setting up governance in your environment structuring your hierarchy and organizing your resources is the first critical step.

In terms of the best practices of setting up governance in your environment structuring your hierarchy and organizing your resources is the first critical step.



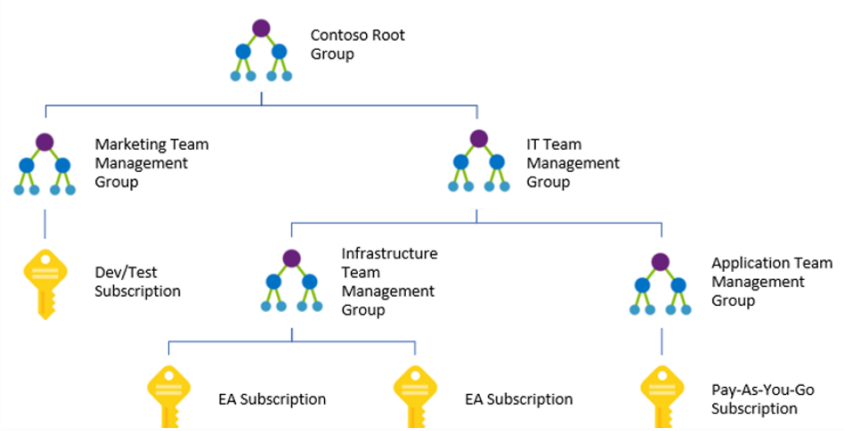
- Creation of customized management hierarchies to suit your organization
- Use of RBAC, tagging policies, cost analysis, and budgets in any scope
- Shared use of policies, security center, and privileged identity management services

CHAPTER 13 CONTINUED

Obviously, in addition to using Management Groups to manage policy enforcement in different environments, it will also assist in organizing your subscriptions.

Another benefit of Management Groups is that if you made the wrong decisions setting up your controls you can create another management group hierarchy and move your subscriptions over without pain.

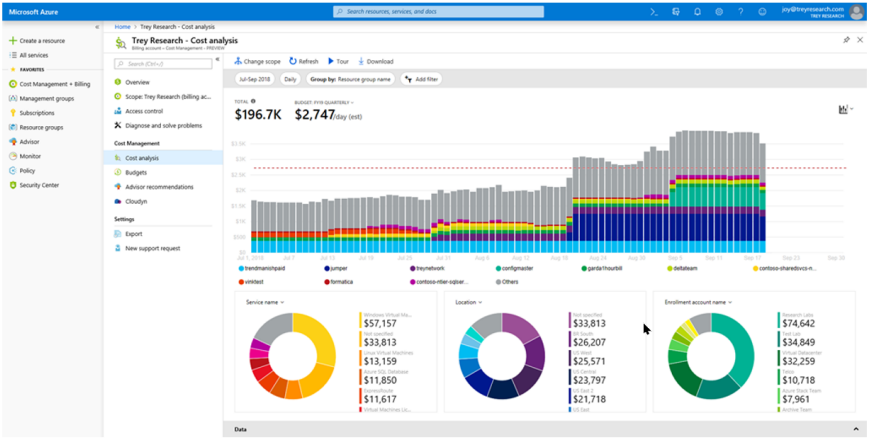
You can also define a more strict set of controls in production env vs test/pre-prod for example. Or you may have a more cost-sensitive control in your test env. Or you can totally isolate your test environment from the Internet…



- What are Azure Management Groups?
- Manage users and groups in the Azure Active Directory

CHAPTER 14

Cost Management



Azure Cost Management is the tool for managing the cost governance of an environment.

It allows you to understand the allocation of costs, create budgets, configure the receipt of alerts based on defined budgets, and the creation of views based on custom criteria. All of these options can be made by different scopes, whether they are Management Groups, Subscriptions, or Resource Groups. It is also possible to download the graphics created in PNG, Excel, and CSV and automatically export the data to a storage account on a daily, weekly, or monthly basis.

Even within Cost Management, you can view Azure Advisor cost recommendations, and access your invoice data and associated payment methods. In addition to this, you can also purchase reservations from Azure, where initially it was only possible to purchase reservations for virtual machines however today it is now possible to reserve resources such as storage, data services, and software plans such as SUSE Linux, Red Hat, and VMWare. In addition to all this, an additional feature that may be interesting if you are in a multi-cloud strategy is to allow you to log in to your AWS account and manage multi-cloud billing through a single tool.

- What is Microsoft Cost Management and Billing?
- Control Azure spending and manage bills with Microsoft Cost Management + Billing

CHAPTER 15

Final Considerations

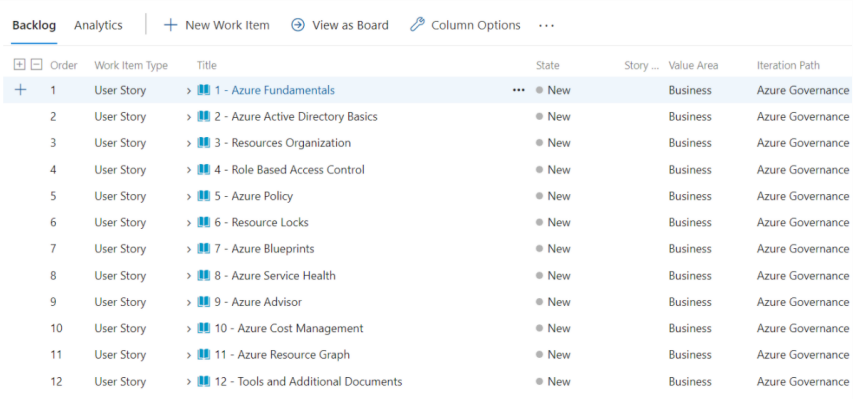
In this final chapter, I would like to share some tools that can be valuable in your Azure governance management.

Azure DevOps Governance Generator

Now that you have the knowledge about the importance of adopting Governance and what tools Azure makes available for you to implement, how about starting to put it into practice?

Having a board containing all the information related to governance from the details of how it works to the details on how to implement it could be useful? Also, if you could share this board with your entire team to discuss each point, delegate activities, create iterations and track the progress of each task, would it be interesting?

So come on. Visit this link and find out how to use Azure DevOps Generator to get it all for free and start implementing Azure Governance in your organization.



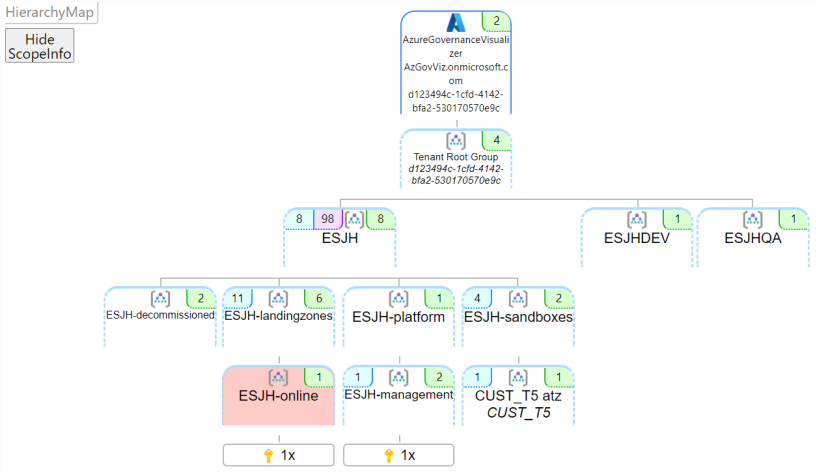
Order	Work Item Type	Title	State	Story ...	Value Area	Iteration Path
1	User Story	1 - Azure Fundamentals	New		Business	Azure Governance
2	User Story	2 - Azure Active Directory Basics	New		Business	Azure Governance
3	User Story	3 - Resources Organization	New		Business	Azure Governance
4	User Story	4 - Role Based Access Control	New		Business	Azure Governance
5	User Story	5 - Azure Policy	New		Business	Azure Governance
6	User Story	6 - Resource Locks	New		Business	Azure Governance
7	User Story	7 - Azure Blueprints	New		Business	Azure Governance
8	User Story	8 - Azure Service Health	New		Business	Azure Governance
9	User Story	9 - Azure Advisor	New		Business	Azure Governance
10	User Story	10 - Azure Cost Management	New		Business	Azure Governance
11	User Story	11 - Azure Resource Graph	New		Business	Azure Governance
12	User Story	12 - Tools and Additional Documents	New		Business	Azure Governance

CHAPTER 15 CONTINUED

Azure Governance Visualizer

What do you think about a have a graphical representation of your Governance implementation? Let me present you with one of my favorite tools: AzGovViz.

The AzGovViz (Azure governance visualizer) is a PowerShell script that iterates through an Azure tenant's management group hierarchy down to the subscription level. It captures data from the most relevant Azure governance capabilities such as Azure Policy, Azure role-based access control (Azure RBAC), and Azure Blueprints. From the collected data, the visualizer shows your hierarchy map, creates a tenant summary, and builds granular scope insights about your management groups and subscriptions.



Azure Workbook for Landing Zone Review

The Landing Zone Workbook is something you can deploy to your environment to validate the usage of all Azure CAF best practices. Is available here and is highly recommended to use.

The aim of this workbook is to visualise core components of an Azure Landing Zone with the focus on the core components. This workbook currently visualises the following checks:

- Governance
  - Subscription health
  - Tag use
  - Policy Assignments
  - Resource Locks use
  - Azure Security Center/Defender status + Secure Score
  - Azure Monitor components + Log Analytics workspaces
- Identity and RBAC
  - Azure Advisor findings around Identity and Access
- Networking
  - Subnets without NSGs
  - Virtual Network Gateways
- Compute
  - Virtual Machines wih public IP addresses directly assigned
  - Virtual Machines with unmanaged disks
- Storage
  - Storage accounts with Secure Transfer Only disabled

PSRule

PSRule for Azure is a pre-built set of tests and documentation to help you configure Azure solutions. These tests allow you to check your Infrastructure as Code (IaC) before or after deployment to Azure. PSRule for Azure includes tests that check how IaC is written and how Azure resources are configured.

See more at https://azure.github.io/PSRule.Rules.Azure/

FINAL CONSIDERATIONS

By implementing robust governance in Azure, you transform the Microsoft cloud into a powerful ally, paving the way for success and innovation. Harness the full potential of the cloud computing platform and steer your organization towards a future of excellence and efficiency.