# AZ-104T00A – Administer Governance and Compliance

Welcome to this second appointment with our AZ-104 course. Last time we discuss Identity in Azure, and in particular we saw Azure AD, including the benefits, we described some basic concepts such as Identity, Azure AD account, Azure Tenant. Then we compared AD DS with Azure AD, and the main differences we discuss were protocols and structure. One has a hierarchical structure, the other has a flat structure since it is a managed service. We saw the 4 different editions with respective license model. We talked about Devices, how to integrate in our hybrid environment with the three options we have in Azure AD. Registering, Joining and Hybrid Joining. Then we saw Users, Groups, Administrative Units and how to use Portal, PowerShell or Command line to create them. Any questions from the last session? I saw only 13 people completed the final assessment. I suggest you to do it, just to better understand whether what I explain is being assimilated or I need to change something. Today we are going to talk about Governance and Compliance in Azure. We’ll start with Subscriptions, then Azure Policy and Role-Based Access Control at the end. As last time I’ll try to answer your questions in the last 10 minutes or through mail.

## Configure Subscriptions

Identify Azure Regions

In our job, we are dealing with IT applications and services, computing, storage and network and even if Azure offers internet-based services, it’s made up of datacentres located around the globe. These datacentres are organized and made available to end users by region. A [region](https://azure.microsoft.com/global-infrastructure/regions/) is a geographical area on the planet containing at least one, but potentially multiple datacentres. The datacentres are in close proximity and networked together with a low-latency network. A few examples of regions are West US, West Europe, Australia East, Japan West, Southeast Asia. Azure is generally available in more than 60 regions in 140 countries and it has more global regions than any other cloud provider. Regions provide us with the flexibility and scale needed to bring applications closer to our users and preserve data residency offering compliance and resiliency options for customers.

Region Pairs

Most Azure regions are paired with another region within the same geography to make a regional pair (or paired regions). Regional pairs help to support always-on availability of Azure resources used by our infrastructure. We need to consider some key factor when using regions or paired regions. Some services or Azure Virtual Machine features are available only in certain regions, such as specific Virtual Machine sizes or storage types. Some global Azure services don’t require you to select a region. These services include Azure Active Directory, Microsoft Azure Traffic Manager and Azure DNS. There are also exceptions to region pairing. Check the Azure website for current region availability and exceptions. If you plan to support the Singapore region, note this region is paired with a region outside its geography. The Brazilian South Region also has an exception to standard regional pairing. We have also to consider benefits of data residency that can help us to meet requirements for tax and law enforcement jurisdiction purposes or as in our case, for financial regulation.

Availability Zones

Inside a region we have the Availability Zones. They are physically separated locations that includes one or more datacenters, equipped with independent power, cooling and networking. It acts as an isolation boundary, if one availability zone goes down, the other continues working taking availability sets to the next level.

Azure Sovereign Regions US

There are also special regions called “Azure Sovereign Regions” that run separated instance of Azure. There is one physically isolated from non-US government deployments. It meets the security and compliance needs of US federal agencies, state and local governments, and their solution providers.

Azure Sovereign Regions China

Another special is for China in which Microsoft provides a physically separated instance of Azure Cloud Services operated by 21Vianet (that is a Chinese carrier-neutral internet and data center service provider).

You can get an overview of the Azure infrastructure at this link: <https://infrastructuremap.microsoft.com/>

Implement Azure Subscriptions

Access to Azure resources and services is obtained through Azure subscriptions which forms the core of an Azure environment. It’s a foundational component of every Azure implementation, in fact every resource that we create in Azure resides in an Azure subscription. We can consider it as billing boundary for Azure resources. On one hand, Azure Administrators must understand how to choose the Azure subscriptions that support the company's business needs, and on the other hand, financial controllers for the company need to know how to manage the costs of the subscription services for the organization. There is a specific Azure Service for costs analysis called Microsoft Cost Management and Billing. It provides support for administrative billing tasks and helps us manage billing access to costs. We can use the product to monitor and control Azure spending, and optimize our Azure resource usage, but is out of scope for this course, but I’ll put some links in Github for your reference. As you can see in this slide, Multiple subscriptions can be linked to the same Azure account. Billing for Azure services is done on a per-subscription basis so we can assign different subscription to different environments or projects to control their budgets. Consider setting up different subscriptions and payment options according to our company's departments, projects, regional offices, and so on. On the other hand, consider also a dedicated shared services subscription to ensure all common network resources are billed together and isolated from other workloads. Examples of shared services subscriptions include Azure ExpressRoute (an Azure network service that let us create private connections between Microsoft Datacentres and on-premises datacentres) and Virtual WAN (another Azure network service that brings many networking, security and routing functionalities together to provide a single operational interface).

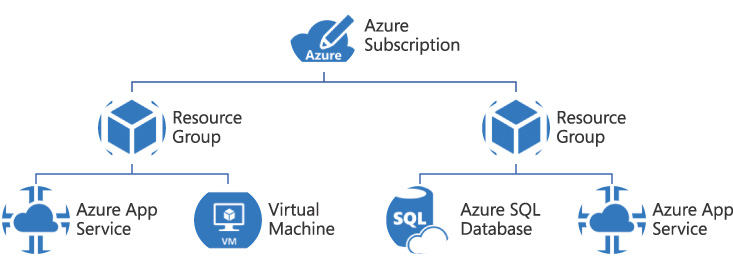
Identify Subscription Usage

There are a wide range of subscription types (or offers), like Free trial, Pay-As-You-Go, Cloud Solution Provider, Enterprise, Student and Visual Studio or MSDN subscription. The capabilities of each subscription are similar, they allow you to create and manage resources. Some subscription types have restrictions on supported resource types and locations. For example, Visual Studio subscriptions typically do not have a credit card associated with them, which prevents you from purchasing services from the Azure Marketplace, such as network virtual appliances.

Obtain a Subscription

We can obtain a Subscription through Enterprise Agreement, through Resellers or Partners or Online with a personal free account to begin

Create Resource Groups

As we build and deploy services in Azure, we will create many types of resources. For instance, when creating our first Virtual Machine, we will also deploy many other resources including a disk for the OS, a Network interface for the VM, a virtual network and subnet for that network interface to bind to and a network security group for example. A resource in Azure is a single service instance, which can be a virtual machine, a storage account, a virtual network or any other Azure service. Resource Groups are logical groupings of resources or those single-service instances. We can group multiple virtual machine instances and manage them as one unit. Resource groups are also very effective for management of costs. A key benefit of using resource groups in Azure is that by grouping related resources together, they typically share a unified lifecycle. Although Azure requires the admin to specify a region when creating a resource group, resources contained within that resource group can span across multiple regions. The requirement of a resource group being deployed to a specific region comes from the need to store the deployment metadata and definitions associated with that resource group in a specific location. As such, it doesn't dictate that resources belonging to that resource group, need to be in the same region as the resource group itself. As easy as resource groups make management of Azure resources, it's important to use resource groups with care. The key to clean and meaningful use of resource groups is an understanding of the resources that are included in them. For example, if a line of business application requires resources that need to be updated together, it makes sense to group these resources in the same resource group. It's equally important, however, to use different resource groups to separate out dev/test resources, staging resources, production resources, et cetera. This is important because resources in each bucket so to speak will typically have different lifecycles. There are other things to consider when creating Resource groups like each resource in Azure can only exists in one resource group. Resource groups cannot be renamed or nested. We can also lock some resources preventing changes or deletion.

Determine Resource Limits

The phrase that best fits the concept of resources created in the public cloud, and which I use often, is: “The cloud is nothing more than someone else's computer”. So, there are no infinite resources. even public cloud resources are subject to physical limitations. We cannot create more virtual machines than a datacentre is capable of hosting without requesting a capacity expansion from the provider in advance. On the subscription there are specific limits on Resources that we can create. Some limits are managed at a regional level. We can ask, opening a support case, an increase of those limits. For example, to request a quota increase with support for vCPU, we must decide how many vCPUs we want to use in which regions. We then request an increase in vCPU quotas for the amounts and regions that we want. If we need to use 24 vCPUs in Singapore to run application there, we specifically request 24 vCPUs in that region. Our vCPU quota is not increased in any other region. Of course, only specific types of subscription can request this change, Free Trial subscription are not eligible for limit or quota increase.

Create an Azure Resource Hierarchy

Azure provides four levels of management: Management groups, subscriptions, resource groups and resources. Management groups help us manage access, policy and compliance for multiple subscriptions. All subscriptions in a management group inherit the conditions that are applied to the management group. So, we can use Management Groups to apply governance consistently across subscriptions, including the application of common Role-based Access controls and the application of Azure Policy that we’ll discuss later on. Management groups allow subscriptions to be organized in a multi-level hierarchy up to six levels deep, excluding the root and subscription levels, providing a number of tangible benefits like **Reducing overhead** because there is no need to apply governance on every subscription. **Enforcement.** Company admins can apply governance at the management group level, outside the control of the subscription admin. The controls implemented at the management group can be applied to both existing and new subscriptions. This eliminates inconsistencies in the application of governance as the same controls are applied the same way to the desired subscriptions. **Reporting.** The Azure Policy provides reports of compliance; with management groups the reporting can span across multiple/all subscriptions in an organization. In this slide there is a single root management group at the root of the hierarchy. This management group is associated with Azure AD tenant that is then associated with an Azure subscription. It cannot be moved or deleted. I’m going to show you in the Azure portal what I defined for my tenant.

Apply Resource Tagging

One of the most important tasks related to management of Azure resources is applying tags. We can apply tags to our Azure resources to logically organize them by categories. Tags are useful for sorting, searching, managing and doing analysis on our resources. Each resource tag consists of a name and a value. We could have the tag name “Environment” and the value “Production” or “Development” and then apply the tag/value pair to our Compute resources. As tags are applied, we can query the resources in our subscription using our tags, and we can even do this across resource groups. This allows us to understand related resources across resource groups for both billing and management. Tags are also included in the billing data for Azure Enterprise Agreement subscriptions through EA Portal and for non-EA subscriptions through the Account Portal. As helpful as tags, there are some limitations that apply to them. For example, each resource or resource groups is limited to a maximum of 50 tag name/value pairs that can be directly applied to the resource group or resource. Another limitation is tag name length. Tag names cannot exceed 512 characters, for storage accounts tag names are limited to 128 characters. VMs cannot exceed 2048 characters for all tag names and values combined. It's important to note that tags that are applied to a resource group are not inherited by the resources within that resource group. Also, tags can't be applied to classic resources such as Cloud Services and they can't contain special characters like % (percent), & (ampersand), \ (backslash), / (slash),? (Question mark), < (less than), > (greater than)

Manage Costs

Since the subscription is our cost boundary, we have to understand which are the factors that can influence it. There are six primary factors affecting costs: **Resource Type**. Costs are resource-specific, so the usage that a meter tracks and the number of meters associated with a resource depend on the resource type. **Services**: Azure usage rates and billing periods can differ between Enterprise, Web Direct, and CSP customers. **Location**: The Azure infrastructure is globally distributed, and usage costs might vary between locations that offer Azure products, services, and resources. **Bandwidth:**Some inbound data transfers are free, such as data going into Azure datacenters. For outbound data transfers, such as data going out of Azure datacenters, pricing is based on Zones. **Azure Reservations:** Azure Reservations allow you to prepay for one-year or three-years of virtual machine or SQL Database compute capacity. Pre-paying will allow you to get a discount on the resources you use. Azure reservations can significantly reduce your virtual machine or SQL database compute costs — up to 72 percent on pay-as-you-go prices with one-year or three-year upfront commitment. Reservations provide a billing discount and don't affect the runtime state of your virtual machines or SQL databases. **Azure Hybrid Benefit:** If you already have Windows Server or SQL Server licenses in your on-premises deployments, you can use the Azure Hybrid Benefit program to save in Azure. With the Windows Server benefit, each license covers the cost of the OS (up to two virtual machines), and you only pay for base compute costs. You can use existing SQL Server licenses to save up to 55 percent on vCore-based SQL Database options. Options include SQL Server in Azure Virtual Machines and SQL Server Integration Services.

## Configure Azure Policy

The next topic regarding governance is about Policy. IT governance is critical to organizations. It ensures achievable goals through effective and efficient use of IT by creating clarity between business goals and IT projects. As such, IT governance requires careful planning of initiatives and setting priorities on a strategic level to help manage and prevent issues. Azure Policy helps accomplish this.

Implement Azure Policies

Azure Policy is an Azure service used to create, assign and manage policies that enforce different rules over Azure resources. In doing so, such resources remain compliant with corporate standards and SLAs. Azure Policy accomplishes this by evaluating deployed resources and scanning for those not compliant with the policies that have been defined. An example of this, would be a case in which a policy is defined to allow only a certain size of virtual machine in an Azure environment. Such a policy, once implemented, would be evaluated when creating and updating resources, as well as over existing resources. Azure policy and Azure RBAC are often used in combination. Where Azure RBAC controls individual user access, group access and rights to our Azure environments at a specific scope, Azure Policy provides a mechanism to express how the environment is governed for all users at a specific scope regardless of any RBAC assignments. Another way to state this is that Azure RBAC is a default deny mechanism with an explicit allow mechanism, whereas Policy is a default allow mechanism with an explicit deny system.

Create Azure Policies

Managing resources with Azure Policy begins with the creation of a policy definition in the portal. Attached to such a definition are conditions under which it is enforced along with an effect that takes place when the defined conditions are met. Azure Policy offers several built-in policies that are available by default. **Azure SQL Database should be running TLS version 1.2 or newer** contains conditions and rules to ensure that all Azure SQL Database deployed use at least TLS version 1.2. This policy denies all servers that do not meet these criteria. **Storage Account should be limited by allowed SKUs** (Stock Keeping Unit – An identification code assigned to each product in the Azure Catalog) contains conditions and rules that determine if a storage account being deployed is within a certain set of SKU sizes. It denies all storage accounts that do not adhere to the defined set of SKU sizes. **Allowed Resource Type** contains conditions and rules to specify which resource types can be deployed. This policy denies any resources that are not part of this defined list. **Not allowed resource types** enables the ability to specify resource types that cannot be deployed. **Allowed Locations** restricts the locations to which resources can be deployed. It is used to enforce geo-compliance requirements. **The Allowed Virtual Machine SKUs** policy restricts the set of virtual machine SKUs that can be deployed. **Require a tag and its default value** applies a required tag and its default value to resources if it is not specified by the user. To leverage these policy definitions as well as any other custom definitions, they need to first be assigned. This can be accomplished through the Azure portal, PowerShell or through Azure CLI. Policy re-evaluation happens about once an hour. As such, changes to a policy definition after implementation of the policy will be re-evaluated over the resources within the hour.

Demonstration – Azure Policy

In this demonstration, we'll walk through the process of creating a policy assignment to restrict the locations to which resources can be deployed, we create also a custom definition and an initiative. To implement Azure Policy, a Policy definition must first be authored. As mentioned, a policy definition describes our desired behaviour for Azure resources at the time resources are created or updated. We can create a Policy that states that resources can only be created in the Switzerland North for an entire subscription. If a user attempts to create a resource in East US, Azure Policy can deny the creation of the resource because it does not meet the stated compliance goal for allowed regions. In this example, Policy is used to deny the creation of a resource and to enforce organizational standards. A couple of slides before we discussed about management hierarchy, and I told you that Policy can also be managed and applied at the management group scope. By associating policies with management groups, Policy definitions and Policy assignments can be shared across multiple subscriptions. This includes the ability to monitor multiple subscriptions for compliance. Now we can create a custom policy definition and we can call “Require VM SKUs not in F series” and in the description we can say “This policy definition enforces that all virtual machines created in this scope have SKUs other than the F series”, we can choose an existing category such as “Compute” and we can change the JSON file with one we compiled before:

{

"policyRule": {

"if": {

"allOf": [{

"field": "type",

"equals": "Microsoft.Compute/virtualMachines"

},

{

"field": "Microsoft.Compute/virtualMachines/sku.name",

"like": "Standard\_F\*"

}

]

},

"then": {

"effect": "deny"

}

}

}

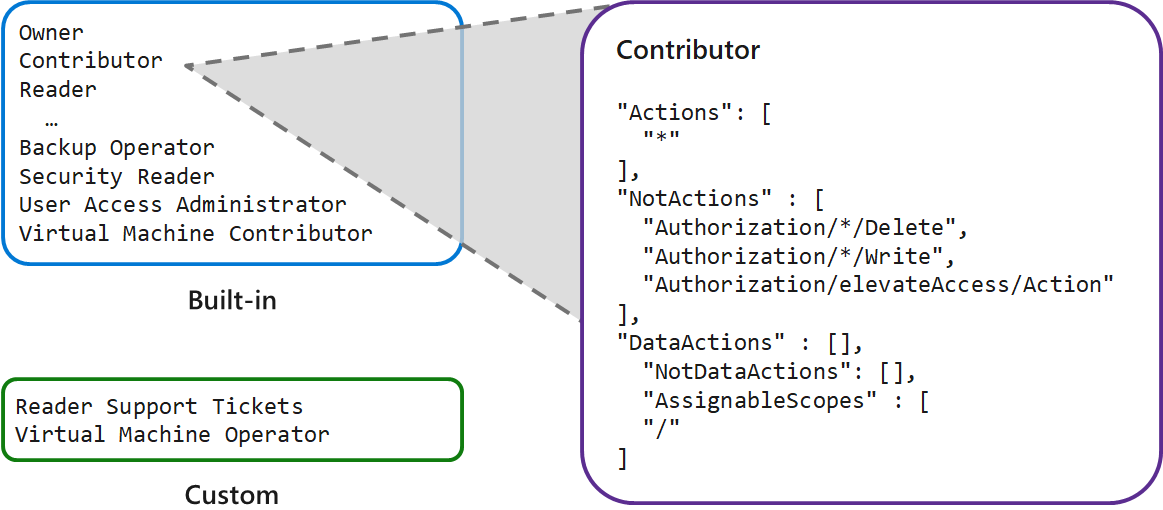
Create a definition with PowerShell:

## $definition = New-AzPolicyDefinition -Name 'denyCoolTiering' -Description 'Deny cool access tiering for storage' -Policy coolAccessTier.json

## Configure Role-Based Access Control

Restricting access based on the need to know and least privilege security principals is how we approach security as an administrator. In Azure, we can leverage role‑based access control to help in this approach. You can use role‑based access control to assign permissions to users, groups, and applications at a certain scope. The scope of a role assignment can be, as we saw for Azure Policy, a management group and all the way down to the resource itself. It provides fine-grained access management to our resources in Azure. It allows us to segregate duties between different teams in our organization. As an example, one team could be tasked with managing VMs and a subscription, or someone could be tasked with cost management, or give our IT security team access to Microsoft Defender for Cloud. You can use the built‑in roles in Azure to assign privileges to users, or, as we will see, you can create a custom role, starting from an existing one.

Create a Role Assignment

In order to better understand RBAC, we need to know the players. First, we'll start out with the security principals. If you're already a Windows Server admin, a few of these should be familiar. We have the user. It's an individual who has a profile in the Azure Active Directory. You can also assign roles to users in other tenants, which is important to understand for the exam. Then we have groups, basically a set of users created in Azure Active Directory. When you assign a role to the group, all users in that group will have that role, much like we do in our on‑prem management. And then we have a service principal. This is a security identity used by applications or services to access specific Azure Resources. You can think of it as a username and password for that application. Then we have a managed identity. This is an identity in Azure Active Directory that is automatically managed by Azure. You typically use the managed identities when developing cloud applications, like managing the credentials for authenticating to Azure services. Azure Key Vault would be an example, rather than storing the credentials in the code. Same with virtual machines. You'll see the identity option under virtual machines. Each resource can have only one system‑assigned identity. Now, as I said, Azure includes built‑in roles based on the services. They also have broad‑based roles, as we will see here. The first is Owner. the Owner has full access to resources, including the right to delegate access to others. As a Windows admin, this sounds a lot like the full control file permission with the ability to grant access. Now, if we drill down a little bit more, there's a built‑in role called Contributor. It can create and manage all types of Azure resources, but can't grant access to others. Now be mindful with these roles as they only apply to the resource itself, not what the resource may contain. So, for instance, a Storage Account Contributor can manage storage accounts, but does not have access to the data within. Again, they can manage the account, but they do not have access to the data. That's an important concept for the exam. And then we have Reader. Now Reader, just like in file permissions, they can view existing Azure resources. So, we might have a Backup Reader, which can only view the backup jobs and settings, but cannot make any changes. Also note, if you have the Reader role assigned to a virtual machine, you're not going to be able to start or stop that virtual machine, only see the settings. Lastly, we have the User Access Administrator role that is granted permission to manage access to Azure resources. For the exam, understand that you will have an Owner, Contributor, Reader and User Access Administrator. For example, there will be a Storage Blob Owner, Storage Blob Contributor, and Storage Blob Reader, so you'll see services with all those different roles. Azure RBAC uses the additive model. As you begin to apply roles to security principals in Azure, it is not uncommon to have overlapping assignments where a security principal is assigned a different role assignment at both a parent and a child scope. For example, if a user is granted Contribute rights at the management group scope and then is granted Reader rights in a subscription, the user will still have Contribute rights across the subscription along with Contribute rights to any other subscriptions under the management group. Another way to think of this is that the most privileged access right takes precedence. Now, along with allowing access, Azure has a feature to explicitly deny access called deny assignments. This is very similar to the Windows deny file permission. This blocks users from performing specific actions, even if a role assignment allows it. So, like I said, very similar to the file permissions in Windows. Denying our permission trumps any allow permission. The final piece to understand with role‑based access control is scope. When you grant access at a parent scope, those permissions are then inherited to the child scopes. For example, if you assign the Owner role to a user at the management groups scope, that user can manage everything in all subscriptions in the management group. Then, if you assign the Reader role to a group at the subscriptions scope, the members of that group can view every resource group and resource in the subscription. Moving forward, if you assign the Contributor role to an application at the resource group scope, it can manage resources of all types in that resource group, but not other resource groups in the subscription. And then finally, and less practical for users and groups, we can assign access to each individual resource.

Create a Role Definition

The specific permissions that are applied to a resource with RBAC are defined in a role definition. A role definition contains the list of permissions, or declared permissions, and those permissions define what actions can or cannot be performed against a type of resource, such as read, write, or delete. Role definitions, or roles, can be either built-in or custom. There are a huge number of built-in role definitions in Azure (over 100). A role definition consists of sets of permissions that are defined in a JSON file. Each permission set has a name, such as Actions or NotActions that describes the purpose of the permissions. This example shows details for the Contributor role in Azure RBAC, which has three sets of permissions. The Actions permissions show the Contributor role has all action privileges. The asterisk “\*” wildcard means “all”. The NotAction permissions narrow the privileges provided by the Actions set, and deny three actions: Authorization/\*/delete – not authorized to delete or remove for “all”, Authorization/\*/write – not authorized to write or change for “all”, Authorization/elevateAccess/Action – not authorized to increase the level or scope of access privileges. The Contributor role also has two DataActions permissions to specify how data can be affected: “NotDataActions”: [] – No specific actions are listed, “AssignableScopes”: [“/”] – The role can be assigned for all scopes that affect data.

Compare Azure RBAC Roles to Azure AD Roles

Azure RBAC roles are more flexible than classic administrator roles and allow for more fine-grained access management. RBAC roles are different from the Azure AD administrative. RBAC roles are used to manage access and allow or restrict users to Azure resources, while Azure AD administrative roles are used to allow or restrict admins to perform identity tasks, such as creating new users, resetting the users’ passwords, and so on. For example, a user who is granted Global Administrator rights in Azure AD does not have permissions to create resources in Azure, but he or she can perform all the identity tasks for an Azure AD tenant.

Apply RBAC Authentication

The following diagram illustrates how you can apply Azure AD administrator roles and Azure RBAC roles in your organization.

Demonstration – Azure RBAC

Create a Resource Group and locate blade “Access control IAM” in the menu. Click on Tab “Roles” and open the role “Owner” to view the permissions. Create a new user and a new group with the option “Azure AD roles can be assigned to the group” checked. Assign the user to the group and open Resource group IAM blades. Assign the role Owner to the Group and check the user permissions.