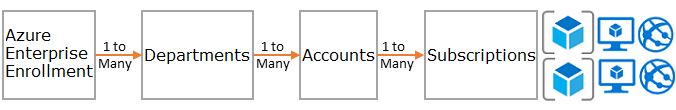
<https://docs.microsoft.com/en-us/azure/architecture/cloud-adoption/appendix/azure-scaffold?toc=/azure/governance/toc.json#define-your-hierarchy>

# Define your hierarchy

The foundation of the scaffold is the hierarchy and relationship of the Azure Enterprise Enrollment through to subscriptions and resource groups. The enterprise enrollment defines the shape and use of Azure services within your company from a contractual point of view. Within the enterprise agreement, you can further subdivide the environment into departments, accounts, and finally, subscriptions and resource groups to match your organization's structure.



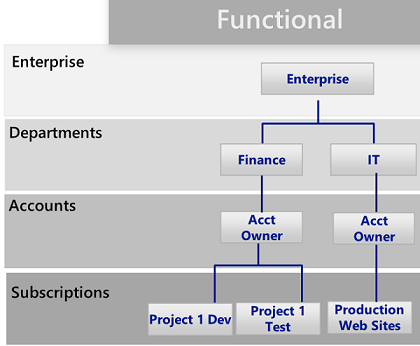
An Azure subscription is the basic unit where all resources are contained. It also defines several limits within Azure, such as number of cores, virtual networks and other resources. Azure Resource Groups are used to further refine the subscription model and enable a more natural grouping of resources.

Every enterprise is different and the hierarchy in the above image allows for significant flexibility in how Azure is organized within your company. Modeling your hierarchy to reflect the needs of your company for billing, resource management, and resource access is the first — and most important — decision you make when starting in the public cloud.

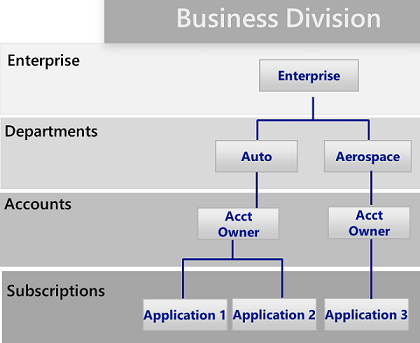
**Departments and Accounts**

The three common patterns for Azure Enrollments are:

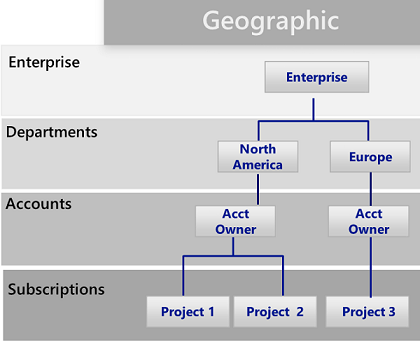
* The **functional** pattern



* The **business unit** pattern



* The **geographic** pattern



# Azure management groups

https://docs.microsoft.com/en-us/azure/governance/management-groups/index

Microsoft has recently released a new way of modeling your hierarchy: Azure management groups. Management groups are much more flexible than departments and accounts and can be nested up to six levels. Management groups allow you to create a hierarchy that is separate from your billing hierarchy, solely for efficient management of resources. Management groups can mirror your billing hierarchy and often enterprises start that way. However, **the power of management groups is when you use them to model your organization where related subscriptions — regardless where they are in the billing hierarchy — are grouped together and need common roles assigned as well as policies and initiatives**. A few examples:

* Production/Non-Production. Some enterprises create management groups to identify their production and non-production subscriptions. Management groups allow these customers to more easily manage roles and policies, for example: non-production subscription may allow developers "contributor" access, but in production, they have only "reader" access.
* Internal Services/External Services. Much like Production/Non-Production, enterprises often have different requirements, policies and roles for internal services versus external (customer facing) services.

Well thought out management groups are, along with Azure Policy and Initiatives the backbone of efficient governance of Azure.

If your organization has many subscriptions, you may need a way to efficiently manage access, policies, and compliance for those subscriptions. 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.

For example, you can apply policies to a management group that limits the regions available for virtual machine (VM) creation. This policy would be applied to all management groups, subscriptions, and resources under that management group by only allowing VMs to be created in that region.

Each directory is given a single top-level management group called the "Root" management group. This root management group is built into the hierarchy to have all management groups and subscriptions fold up to it**. This Root management group allows for global policies and RBAC assignments to be applied at the directory level. The Azure AD Global Administrator needs to elevate themselves to be the owner of this root group initially**. Once the administrator is the owner of the group, they can assign any RBAC role to other directory users or groups to manage the hierarchy.

# Subscriptions

When deciding on your Departments and Accounts (or management groups), you are primarily looking at how you're dividing up your Azure environment to match your organization. Subscriptions, however, are where the real work happens and your decisions here affect security, scalability and billing. Many organizations look at the following patterns as their guides:

* Application/Service: Subscriptions represent an application or a service (portfolio of applications)
* Lifecycle: Subscriptions represent a lifecycle of a service, such as Production or Development.
* Department: Subscriptions represent departments in the organization.

The first two patterns are the most commonly used, and both are highly recommended. The Lifecycle approach is appropriate for most organizations. In this case, the general recommendation is to use two base subscriptions. "Production" and "Non-Production," and then use resource groups to break out the environments further.

# Resource groups

Azure Resource Manager enables you to put resources into meaningful groups for management, billing, or natural affinity. Resource groups are containers of resources that have a common life cycle or share an attribute such as "all SQL servers" or "Application A".

Resource groups can't be nested, and resources can only belong to one resource group. Some actions can act on all resources in a resource group. For example, deleting a resource group removes all resources within the resource group. Like subscriptions, there are common patterns when creating resource groups and will vary from "Traditional IT" workloads to "Agile IT" workloads:

"Traditional IT" workloads are most commonly grouped by items within the same life cycle, such as an application. Grouping by application allows for individual application management.

"Agile IT" workloads tend to focus on external customer-facing cloud applications. The resource groups often reflect the layers of deployment (such as a web tier or app tier) and management.

# Naming standards

The first pillar of the scaffold is a consistent naming standard. Well-designed naming standards enable you to identify resources in the portal, on a bill, and within scripts. You likely already have existing naming standards for on-premises infrastructure. When adding Azure to your environment, you should extend those naming standards to your Azure resources.

For naming conventions:

Review and adopt where possible the Patterns and Practices guidance. This guidance helps you decide on a meaningful naming standard and provides extensive examples.

Using Resource Manager Policies to help enforce naming standards

Remember that it's difficult to change names later, so a few minutes now will save you trouble later.

Concentrate your naming standards on those resources that are more commonly used and searched for. For example, all resource groups should follow a strong standard for clarity.

# Resource Tags

Resource tags are tightly aligned with naming standards. As resources are added to subscriptions, it becomes increasingly important to logically categorize them for billing, management, and operational purposes. For more information, see Use tags to organize your Azure resources.

Tags are used in many ways beyond billing and management. They are often used as part of automation (see later section). This can cause conflicts if not considered up front. The recommended practice is to identify all the common tags at the enterprise level (such as ApplicationOwner, CostCenter) and apply them consistently when deploying resources using automation.

# Azure Policy and initiatives

The second pillar of the scaffold involves using Azure Policy and initiatives to manage risk by enforcing rules (with effects) over the resources and services in your subscriptions. Azure Initiatives are collections of policies that are designed to achieve a single goal. Azure policies and initiatives are then assigned to a resource scope to begin enforcement of the particular policies.

Policies and initiatives are even more powerful when used with the management groups mentioned earlier. Management groups enable the assignment of an initiative or policy to an entire set of subscriptions.

## Common uses of Resource Manager policies

Azure policies and initiatives are a powerful tool in the Azure toolkit. Policies allow companies to provide controls for "Traditional IT" workloads that enable the stability that is needed for line-of-business applications while also allowing "Agile" workloads; such as, developing customer applications without opening up the enterprise to additional risk. The most common patterns we see for policies are:

Geo-compliance/data sovereignty. Azure has an ever-growing list of regions across the world. Enterprises often need to ensure that resources in a particular scope remain in a geographic region to address regulatory requirements.

Avoid exposing servers publicly. Azure Policy can prohibit the deployment of certain resource types. It's common to create a policy to deny the creation of a public IP within a particular scope, avoiding un-intended exposure of a server to the internet.

Cost Management and Metadata. Resource tags are often used to add important billing data to resources and resource groups such as CostCenter, Owner and more. These tags are invaluable for accurate billing and management of resources. Policies can enforce the application of resources tags to all deployed resource, making it easier to manage.

## Common uses of initiatives

The introduction of initiatives provided enterprises a way to group logical policies together and track as a whole. Initiatives further support the enterprise to address the needs of both "agile" and "traditional" workloads. We have seen very creative uses of initiatives, but commonly we see:

* Enable monitoring in Azure Security Center. This is a default initiative in the Azure Policy and an excellent example of what initiative are. It enables policies that identify un-encrypted SQL databases, virtual machine (VM) vulnerabilities and more common security related needs.
* Regulatory specific initiative. Enterprises often group policies common to a regulatory requirement (such as HIPAA) so that controls and compliancy to those controls are tracked efficiently.
* Resource Types and SKUs. Creating an initiative that restricts the types of resources that can be deployed as well as the SKUs that can be deployed can help to control costs and ensure your organization is only deploying resources that your team have the skillset and procedures to support.

We recommend you always use initiative definitions instead of policy definitions. After assigning an initiative to a scope, such as subscription or management group, you can easily add another policy to the initiative without having to change any assignments. This makes understanding what is applied and tracking compliance far easier.

### Policy and Initiative assignments

After the creation of policies and grouping them into logical initiatives you must assign the policy to a scope, whether it is a management group, a subscription or even a resource group. Assignments allow you to also exclude a sub-scope from the assignment of a policy. For example, if you deny the creation of public IPs within a subscription, you could create an assignment with an exclusion for a resource group connected to your protected DMZ.

https://github.com/Azure/azure-policy

# Identity and access management

One of the first, and most crucial, questions you ask yourself when starting with the public cloud is "who should have access to resources?" and "how do I control this access?" Allowing or disallowing access to the Azure portal, and controlling access to resources in the portal is critical to the long term success and safety of your assets in the cloud.

When implementing role-based access, the following are highly recommended:

* Control the Administrator/Co-Administrator of a subscription as these roles have extensive permissions. You only need to add the Subscription Owner as a Co-administrator if they need to managed Azure Classic deployments.
* Use management groups to assign roles across multiple subscriptions and reduce the burden of managing them at the subscription level.
* Add Azure users to a group (for example, Application X Owners) in Active Directory. Use the synced group to provide group members the appropriate rights to manage the resource group containing the application.
* Follow the principle of granting the least privilege required to do the expected work.