Microsoft Azure Governance



www.ngahr.com



Document Control

Document Information

|  |  |
| --- | --- |
| Document Id |  |
| Document Owner | José Enrique Pons |
| Issue Date | 23/03/2017 |
| Last Saved Date | 11 May 2017 |
| File Name |  |

Document History

|  |  |  |
| --- | --- | --- |
| Version | Issue Date | Changes |
| [1.0] | 23/03/17 | First draft. |
|  |  |  |
|  |  |  |
|  |  |  |

Document Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature | Date |
| Project Sponsor | Stuart Curley |  |  |
| Project Sponsor | Robin Haughan |  |  |
| Project Sponsor | Surya Manepalli |  |  |
| Project Review Group | Selvam Jayabalan  David AK Smith  Nick Pratt  Pilar Guillen |  |  |
| Project Manager |  |  |  |
| Quality Manager  (if applicable) |  |  |  |
| Procurement Manager  (if applicable) |  |  |  |
| Communications Manager  (if applicable) |  |  |  |
| Project Office Manager  (if applicable) |  |  |  |

Table of Contents

[1 Document Control 2](#_Toc482175176)

[3 Introduction 4](#_Toc482175177)

[4 Executive Summary 5](#_Toc482175178)

[5 Azure Scaffold 6](#_Toc482175179)

[5.1 Hierarchy definition 6](#_Toc482175180)

[5.1.1 Naming Standards 6](#_Toc482175181)

[5.1.2 Policies and auditing 7](#_Toc482175182)

[5.1.3 Resource tags 8](#_Toc482175183)

[5.1.4 Resource Group 8](#_Toc482175184)

[5.1.5 Role-based access control 8](#_Toc482175185)

[5.1.6 Azure resource locks 8](#_Toc482175186)

[5.1.7 Core networking resources 9](#_Toc482175187)

[5.1.8 Azure Security Center 9](#_Toc482175188)

[6 Azure Governance for NGA 11](#_Toc482175189)

[7 Lifecycle 14](#_Toc482175190)

[7.1 Business Requirements 14](#_Toc482175191)

[7.2 Specification of Azure Requirements 15](#_Toc482175192)

[7.3 Approval of Provision 16](#_Toc482175193)

[7.4 Verification of Provision 16](#_Toc482175194)

[7.5 Run and Operate 16](#_Toc482175195)

[7.6 Automatic Deprovision 16](#_Toc482175196)

[7.7 Decommission 17](#_Toc482175197)

[8 Resources 18](#_Toc482175198)

# Executive Summary

This document studies the architecture of Azure for governance.

1. The key concepts are explained - Defining roles, groups, resource tags and locks.
2. An example with the configuration of an NGA application is also provided.
3. A typical web-based application is considered
4. The typical structure at NGA is considered (Systems teams, Platform operation teams, Application and deployment teams).

Questions are commonly asked about Azure and NGA HR need answers to these too:

* "How do I meet our legal requirements for data sovereignty in certain countries?"
* "How do I ensure that someone does not inadvertently change a critical system?"
* "How do I know what every resource is supporting so I can account for it and bill it back accurately?"

The **Azure Scaffold** process is commented and a procedure to implement it within NGA would be defined. First the possibilities and the configuration provided by Azure are commented. Next, the way to implement will be commented by defining the NGA particular scenario.

The scaffold guides the general outline, and provides anchor points for more permanent systems to be mounted. An enterprise scaffold is the same: a set of flexible controls and Azure capabilities that provide structure to the environment, and anchors for services built on the public cloud. It provides the builders (IT and business groups) a foundation to create and attach new services.

It seems clear that Azure provides an easy process to scaffold the governance. Nevertheless, the way to automatically shut down VM based on the bill is not provided by default. It is essential that NGA has a process for deprovisioning efficiently as much as we need a process for efficient provisioning. This document proposes approaches for both.

MS Azure provides scripting facilities to implement scripts to provide any desired functionality, like ‘politely’ shut down VM and save the work in storage place. Nevertheless, those scripts must be implemented by NGA to fit our needs.

Therefore, we believe that Global Technology should initiate a project to define **Azure Service Governance** in detail, defining all the processes and controls that need to be in place to:

1. Give NGA HR colleagues the best experience possible from Azure Services
2. Ensure that we efficiently and effectively exploit the value of the Azure Services

# Azure Services

## What does Azure offer?

According to Microsoft’s information, Azure is a collection of integrated cloud services to build, deploy and manage applications through Microsoft global network. Besides that, Azure claims to be the most comprehensive compliance coverage cloud with respect to the rest of providers (more certifications that any other provider).

## What services does Azure offer[[1]](#footnote-1)?

Digital Marketing: Worldwide digital campaigns that are personalized and scalable.

Mobile: Support for back ends for mobile apps.

• Notification Hubs: Send push notifications to any platform from any backend.

• Traffic Manager: route incoming traffic for high performance and availability.

• Xamarin Test Cloud: Automate the app testing on thousands of devices in the cloud.

• HockeyApp: Deployment, usage statistics and crash reports for apps.

• Azure Cosmos DB: Distributed multi-model database.

Microservice applications: scalable and reliable applications.

* Container Service: Scale and orchestrate containers using Kubernetes DC/OS or Docker Swarm.
* Service Fabric: Develop microservices and orchestrate containers on Windows or Linux.

Business intelligence services:

* Power BI: Embed interactive data visualizations in applications.
* SQL Data Warehouse: Data warehouse as a service.
* Azure Analysis Services: Combine data from different sources and allow scale up and down applications.

Big data and analytics:

* HDInsight: Provision cloud Hadoop, Spark, R Server, HBase and Storm clusters.
* Data lake analytics: Distributed analytics service that makes big data easy.
* Machine Learning (Studio): A suite for basic ML tasks.

Cloud migration: Move from on premise to cloud.

* Site recovery: Orchestrate protection and recovery of private clouds.
* Advisor: Personalized Azure best practices recommendation engine.
* Log Analytics: collect, search and visualize machine data from on-premise and cloud.

Data warehouse:

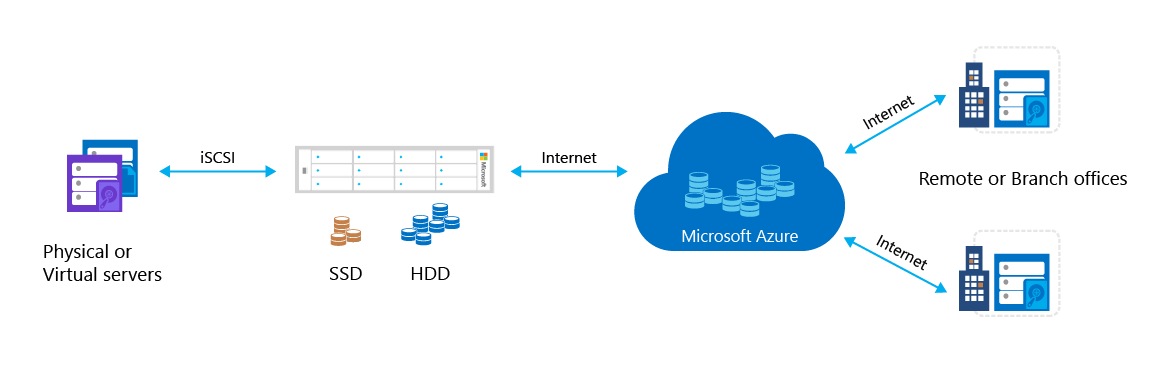
* SQL Data Warehouse.
* Azure Analysis services.
* Data Factory: Orchestrate and manage data transformation and movement.

Business Software as a Service apps:

* SQL databases and Power BI.
* Cognitive services (MS Language Understanding services).
* Azure Active Directory: Enables single sign-on.
* App Service: Deploy web apps in linux containers.

Backup and archive:

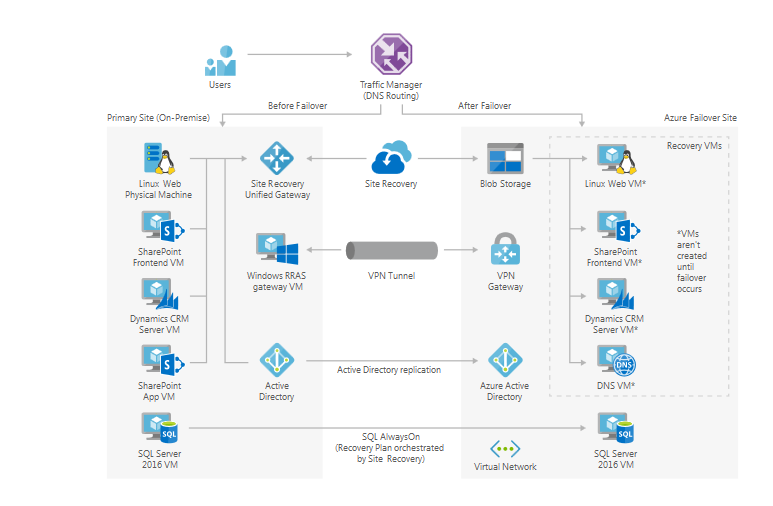
* Backup, Site recovery (enables disaster recovery as a service), Storage and StorSimple services.



Development and test: Simplify the process of building and testing applications across every platform.

* Azure DevTest Labs: Create environments using reusable templates and artifacts.
* Visual Studio Team services: Share code, track work and ship software.
* Application Insights: Detect, triage and diagnose issues in your web apps and services.

Disaster recovery: Disaster recovery as a service and standardized templates:



SAP on Azure, Blockchain support and Line of Business applications.

# Azure Lifecycle

This section is devoted to describe the full lifecycle process starting in the definition of the business requirements, following by the specification of the needed resources and the approval process. Then, the provision, verification and the automatic de-provision procedures are explicated too.

The process of requesting resources from Azure starts by defining the business requirements.

It is essential that NGA HR has clear guidelines and processes for:

* How can NGA people access and use these services?
* What are we putting in place to make these services available to NGA?
* What are the pre-requisites for accessing and using the Azure Services?

## Business Requirements

To understand the resources needed from Azure, the Business Requirements should be specified. The specification of the kind of resources needed (from the Azure catalogue) and the amount of them would be a process. First, the non-functional requirements[[2]](#footnote-2) need to be understood. The performance, security and audit requirements are specified in this phase.

The input to this assessment stage is a form in which the following elements are defined:

* Type of resources from Azure catalogue.
* Non-functional requirements:
  + Security and audit.
  + Backup policies (Maybe to specify a standard would help here).
  + Performance required and scale-up possibilities.

The non-functional requirements should be standardized across NGA as a policy. I.e. for development systems: backups are done on a daily basis but retention policy is 30 days. In production systems, backups are done hourly.

To deploy serverless[[3]](#footnote-3) services (Azure functions[[4]](#footnote-4)), a different approach is needed. A serverless service is basically a way of abstraction in which the infrastructure required to execute a service is not managed by the owner of the service. In other words, the code is deployed to the cloud provider and the infrastructure is managed seamlessly to satisfy all kind of loads. The main benefit for this is the isolation from the underlying infrastructure. This model is based on a budget, which limits the number of calls to the service implemented. In this case, the estimation on the number of calls would be required.

The output of this phase is a form with the estimation of Azure requirements, the timeframe and / or the budget limit.

## Specification of Azure Requirements

This phase consists on the specification of the amount and the name of the requirements defined in the previous stage. The work here is to translate from the pure Azure requirements to all of the following elements required by the governance process:

* Subscriptions.
* Resource groups.
* Policies:
  + Resource tags.
  + Core Network.
  + Resource locks.

To automate this process, a template would be useful. It will generate the required elements (subscriptions, resource groups, policies) from the required Azure components.

The output of this phase is a document with the definition of all requirements needed by Azure. This process may be automated but need to be carefully reviewed to ensure the consistency across the governance process.

## Approval of Provision

The inputs of this phase are two documents: The governance document and the business requirements. If the approval is going to be handled by Service Now, the process analogous to the Standard Changes would apply.

## Verification of Provision

This stage consists on the verification that the approved resources and the approved resources satisfy the business needs. For instance, imagine that only some of the requested resources are approved. Then, this process will ensure that the approved resources will satisfy the requirements.

## Run and Operate

The input of this stage is the approved resources and the approved governance requirements. Then, an automated process will attach the resources to already created ones or will create empty resources when needed.

## Automatic Deprovisioning

Once the time frame or budget requirements are met, the resources would be decommissioned. The first step is to send an email to stakeholders once the time frame is going to its end. Or when the budget is 90% consumed, for instance.

A script would deal with this task. The details for the implementation would be detailed later. Some points to take into account are:

* Production environments can’t be stopped by using this script. Only by using a decommission process should stop these kind of systems.
* A very important point, especially when dealing with data applications, is to avoid the loss of data when stopping the computing resources. Therefore, storage and compute capabilities have to be split.

## Decommissioning

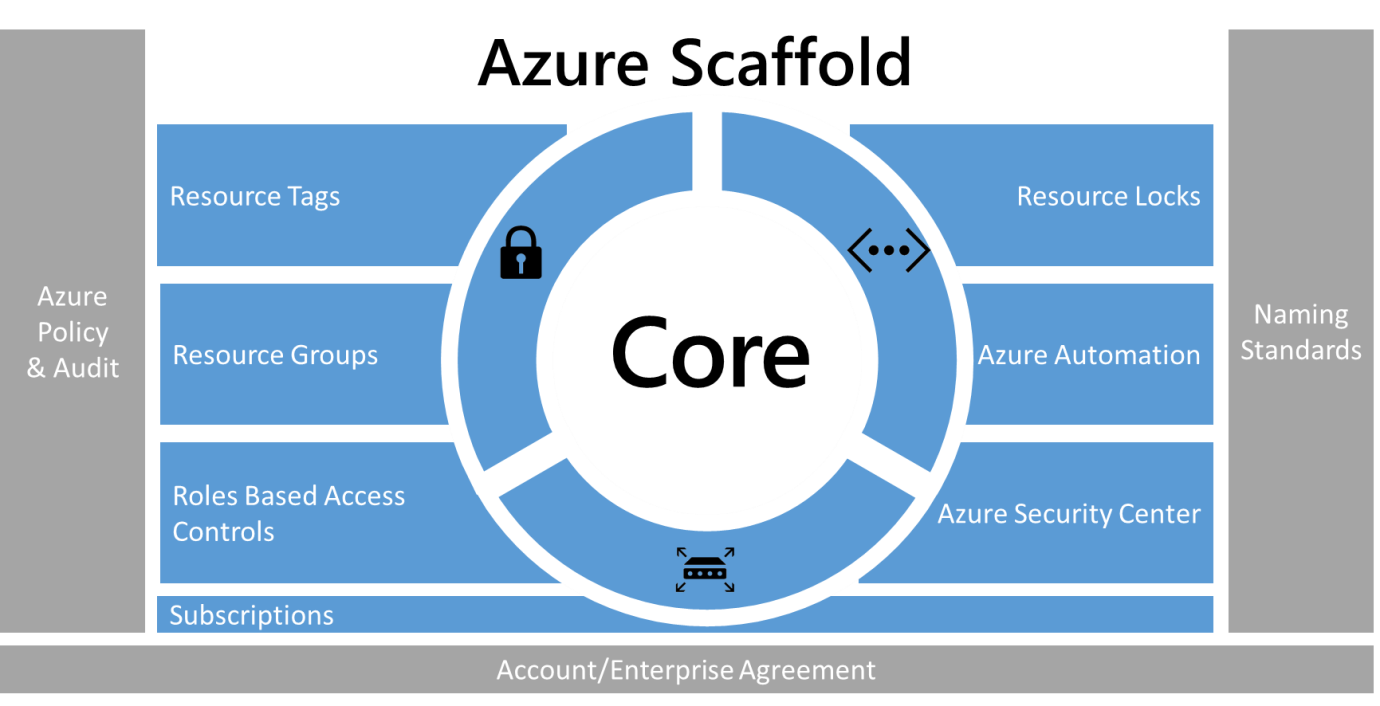
The decommission process is the explicit way of shutting down all Azure resources owned by a project or customer. This should be the only process to stop production instances. The decommission process should be handled in a Snow change normal ticket (with the full approval workflow).

The steps to decommission an Azure resource group are:

1. Request decommission, via Snow Change.
2. Identification of the resources in Azure.
3. Approval of the decommission process.
4. Decommission the Azure resources.
5. Verification that the resources were decommissioned correctly.

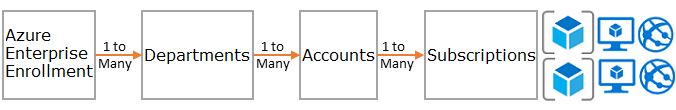
# Azure Scaffold

The following image describes the components of the scaffold. The foundation relies on a solid plan for departments, accounts, and subscriptions. The pillars consist of Resource Manager policies and strong naming standards. The rest of scaffold comes from core Azure capabilities and features that enable a secure and manageable environment.



## Hierarchy definition

The foundation of the scaffold is the Azure Enterprise Enrolment (and the Enterprise Portal). The enterprise enrolment defines the shape and use of Azure services within a company and is the core governance structure. Within the enterprise agreement, customers are able to further subdivide the environment into departments, accounts, and finally, subscriptions. An Azure subscription is the basic unit where all resources are contained. It also defines several limits within Azure, such as number of cores, resources, etc.



Different patterns can be implemented: By function, by business unit or by geographic location.

### Naming Standards

The first pillar of the scaffold is naming standards[[5]](#footnote-5). Well-designed naming standards enable you to identify resources in the portal, on a bill, and within scripts. Most likely, you already have naming standards for on-premise infrastructure. When adding Azure to your environment, you should extend those naming standards to your Azure resources. Naming standard facilitate more efficient management of the environment at all levels.

### Policies and auditing

Resource Manager policies [[6]](#footnote-6)provide you with the ability to manage risk in Azure. You can define policies that ensure data sovereignty by restricting, enforcing, or auditing certain actions.

* Policy is a default allow system. You control actions by defining and assigning policies to resources that deny or audit actions on resources.
* Policies are described by policy definitions in a policy definition language (if-then conditions).
* You create polices with JSON (Javascript Object Notation) formatted files. After defining a policy, you assign it to a particular scope: subscription, resource group, or resource.

Policies have multiple actions that allow for a fine-grained approach to your scenarios. The actions are:

* Deny: Blocks the resource request
* Audit: Allows the request but adds a line to the activity log (which can be used to provide alerts or to trigger runbooks)
* Append: Adds specified information to the resource. For example, if there is not a "CostCenter" tag on a resource, add that tag with a default value.

#### Common uses of Resource Manager policies

Azure Resource Manager policies are a powerful tool in the Azure toolkit. They enable you to avoid unexpected costs, to identify a cost center for resources through tagging, and to ensure that compliancy requirements are met. When policies are combined with the built-in auditing features, you can fashion complex and flexible solutions. Policies allow companies to provide controls for "Traditional IT" workloads and "Agile" workloads; such as, developing customer applications. The most common patterns we see for policies are:

* **Geo-compliance/data sovereignty** - Azure provides regions across the world. Enterprises often wish to control where resources are created (whether to ensure data sovereignty or just to ensure resources are created close to the end consumers of the resources).
* **Cost management** - An Azure subscription can contain resources of many types and scale. Corporations often wish to ensure that standard subscriptions avoid using unnecessarily large resources, which can cost hundreds of dollars a month or more.
* **Default governance through required tags** - Requiring tags is one of the most common and highly desired features. Using Azure Resource Manager Policies enterprises are able to ensure that a resource is appropriately tagged. The most common tags are: Department, Resource Owner, and Environment type (for example - production, test, development)

#### Audit – what happened?

To view how your environment is functioning, you need to audit user activity. Most resource types within Azure create diagnostic logs that you can analyze through a log tool or in Azure Operations Management Suite. You can gather activity logs across multiple subscriptions to provide a departmental or enterprise view. Audit records are both an important diagnostic tool and a crucial mechanism to trigger events in the Azure environment.

Activity logs from Resource Manager deployments enable you to determine the operations that took place and who performed them. Activity logs can be collected and aggregated using tools like Log Analytics.

### Resource tags

Tags are metadata that can be attached to the resources. Some common tags are:

* BillTo.
* Department, Business Unit.
* Environment.
* Application Owner.
* Project Name.

### Resource Group

Resource Manager enables to put resources into meaningful groups for management, billing, or natural affinity. As mentioned earlier, Azure has two deployment models. In the earlier Classic model, the basic unit of management was the subscription. It was difficult to break down resources within a subscription, which led to the creation of large numbers of subscriptions. With the Resource Manager model, we saw the introduction of resource groups. Resource groups are containers of resources that have a common lifecycle or share an attribute such as "all SQL servers" or "Application A".

Resource groups cannot be contained within each other and resources can only belong to one resource group. You can apply certain actions on all resources in a resource group. For example, deleting a resource group removes all resources within the resource group. Typically, you place an entire application or related system in the same resource group. For example, a three tier application called Contoso Web Application would contain the web server, application server and SQL server in the same resource group.

### Role-based access control

You probably are asking yourself "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 crucial.

When Azure was initially released, access controls to a subscription were basic: Administrator or Co-Administrator. Access to a subscription in the Classic model implied access to all the resources in the portal. This lack of fine-grained control led to the proliferation of subscriptions to provide a level of reasonable access control for an Azure Enrollment.

This proliferation of subscriptions is no longer needed. With role-based access control, you can assign users to standard roles (such as common "reader" and "writer" types of roles). You can also define custom roles.

### Azure resource locks

As your organization adds core services to the subscription, it becomes increasingly important to ensure that those services are available to avoid business disruption. Resource locks enable you to restrict operations on high-value resources where modifying or deleting them would have a significant impact on your applications or cloud infrastructure. You can apply locks to a subscription, resource group, or resource. Typically, you apply locks to foundational resources such as virtual networks, gateways, and storage accounts.

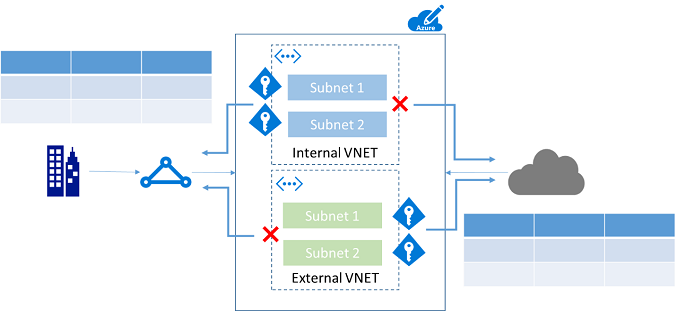
Resource locks currently support two values: CanNotDelete and ReadOnly. CanNotDelete means that users (with the appropriate rights) can still read or modify a resource but cannot delete it. ReadOnly means that authorized users can't delete or modify a resource.

To create or delete management locks, you must have access to Microsoft.Authorization/\* or Microsoft.Authorization/locks/\* actions. Of the built-in roles, only Owner and User Access Administrator are granted those actions.

### Core networking resources

Access to resources can be either internal (within the corporation's network) or external (through the internet). It is easy for users in your organization to inadvertently put resources in the wrong spot, and potentially open them to malicious access. As with on-premise devices, enterprises must add appropriate controls to ensure that Azure users make the right decisions. For subscription governance, we identify core resources that provide basic control of access. The core resources consist of:

* **Virtual networks** are container objects for subnets. Though not strictly necessary, it is often used when connecting applications to internal corporate resources.
* **Network security groups** are similar to a firewall and provide rules for how a resource can "talk" over the network. They provide granular control over how/if a subnet (or virtual machine) can connect to the Internet or other subnets in the same virtual network.



#### Automation

Managing resources individually is both time-consuming and potentially error prone for certain operations. Azure provides various automation capabilities including Azure Automation[[7]](#footnote-7), Logic Apps, and Azure Functions. Azure Automation enables administrators to create and define runbooks to handle common tasks in managing resources. You create runbooks by using either a PowerShell code editor or a graphical editor. You can produce complex multi-stage workflows. Azure Automation is often used to handle common tasks such as shutting down unused resources, or creating resources in response to a specific trigger without needing human intervention.

### Azure Security Center

The Azure Security Center [[8]](#footnote-8)provides a central view of the security status of resources in the subscriptions, and provides recommendations that help prevent compromised resources. It can enable more granular policies (for example, applying policies to specific resource groups that allow the enterprise to tailor their posture to the risk they are addressing). Finally, Azure Security Center is an open platform that enables Microsoft partners and independent software vendors to create software that plugs into Azure Security Center to enhance its capabilities.

# Azure Governance for NGA

This section proposes a model to implement Azure governance within NGA current structure. The line-of-business approach seems to be the most accurate. This approach is based in the infrastructure and the model we have in EMEA region:

* VMs are owned by Systems Operations team.
* Applications are owned by different applications teams.
* Developers access the applications.
* Deployment team do the deployment of new versions of applications.

## Subscription

Taking into account this structure, the following naming standards and resource groups:

|  |  |  |
| --- | --- | --- |
| Item | Name | Description |
| Subscription | NGA x Production | Subscription for the productive application x in production. |
| Resource Group | rgVMProd | Contains the database servers and the application vm. |
| Resource Group | rgCoreNetworks | Virtual network and site-to-site gateway connection. |

## Role-based access control:

Once we have created the subscription, we have to define the Roles for the access control. The easiest way to implement this is by syncing the actual data from NGA Active Directory to Azure Active Directory.

|  |  |  |
| --- | --- | --- |
| Role | Assigned to | Description |
| Owner | Managed ID from NGA AD | This is controlled by using NGA AD and ensures that subscription owner access is fully audited. |
| Security Manager | Security and Risk management department | This role allows users to look at the Azure Security Center and the status of the resources. |
| Network Contributor | Networking Operations Team | Allows the management of the VPN and configuration of the Virtual Networks. |
| (Custom role)  DBA admin | DBA team | This role allows DBA to perform their administration tasks in the database. |
| (Custom role)  Apps team | Application team | This role allows the Application team to access the vms for management and application related tasks. |
| (Custom role)  Development team | Development team | This role allows developers to access the server to do very specific tasks related to development. |
| (Custom role)  Deployment team | Deployment team | This allows the deployment team to use their tools for deployment. |

## Policies:

Taking into account the shape of the landscape in NGA, we can define policies by environment: For instance:

* In production and quality environments, the creation and deletion of resources must be restricted.
* Development / Testing environments may be opened to create resources.
* Internal systems can be billed to their respective teams by using tags.

The following Resource Manager policies can be created:

|  |  |  |
| --- | --- | --- |
| Field | Effect | Description |
| Location | Audit | Audit the creation of resources in any region. |
| type | deny | Deny the creation of expensive VM. |
| tags | deny | Require application owner tag |
| tags | Deny | Require cost center tag |
| Tags | Append | Append tag name BusinessUnit to all resources. |

## Resource tags

The following tasks can be defined for the resource groups and resources:

|  |  |
| --- | --- |
| Tag Name | Tag Value |
| Application Owner | The name of the team that manages the app. |
| CostCenter | The cost center of the group that is paying for the Azure consumption. |
| BusinessUnit | Name of the business unit associated with the subscription. |

## Core Network

If the application faces the public network, then two virtual networks should be defined. Also a security group will act as firewall, by minimizing the surface in which the application lives.

|  |  |  |
| --- | --- | --- |
| Resource type | Name | Description |
| Virtual Network | vnInternal | This connects the application with NGA internal network. |
| Virtual Network | vnExternal | Available for public-facing endpoints. |
| Network Security Group | nsgVMProd | Defines a surface for allow connections only from the subnet where the application is. |

## Resource locks

This is useful to prevent accidental deletion of a production VM. The following resource lock can be defined:

|  |  |  |
| --- | --- | --- |
| Lock type | Resource | Description |
| CanNotDelete | vnInternal | Prevent users from deleting virtual networks or subnets. |

Finally, the configuration of Azure Security Center should be enabled to quickly identify and handle treats.

# Resources

The following resources were accessed to elaborate this document:

|  |  |
| --- | --- |
| Name | Resource |
| Azure Enterprise Scaffold – Prescriptive Subscription Governance | <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-subscription-governance> |
| Examples of Implementing Azure Enterprise Scaffold | <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-subscription-examples> |
| Manage Azure Resources with Resource Policies | <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-policy> |
| View Activity Logs to Audit Actions on Resources | <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-audit> |
| View and Analyse Azure Audit Logs in Power BI | <https://azure.microsoft.com/en-us/blog/analyze-azure-audit-logs-in-powerbi-more/> |
| Configuration Steps: | <https://powerbi.microsoft.com/en-us/documentation/powerbi-content-pack-azure-audit-logs/> |
| Prevent Changes to Azure Resources using Resource Locks | <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-lock-resources> |
| Manage Access to Azure Resources and Azure Management Portal using AAD and RBAC | <https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-what-is> |
| Azure RBAC built in Roles and Responsibilities | <https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-built-in-roles> |

1. <https://azure.microsoft.com/en-us/solutions/> [↑](#footnote-ref-1)
2. <https://en.wikipedia.org/wiki/Non-functional_requirement> [↑](#footnote-ref-2)
3. <https://serverless.com/> [↑](#footnote-ref-3)
4. <https://azure.microsoft.com/en-us/services/functions/> [↑](#footnote-ref-4)
5. See: <https://docs.microsoft.com/en-us/azure/architecture/best-practices/naming-conventions> [↑](#footnote-ref-5)
6. See: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-policy> [↑](#footnote-ref-6)
7. See: <https://docs.microsoft.com/en-us/azure/automation/automation-intro> [↑](#footnote-ref-7)
8. See: <https://docs.microsoft.com/en-us/azure/security-center/security-center-intro> [↑](#footnote-ref-8)