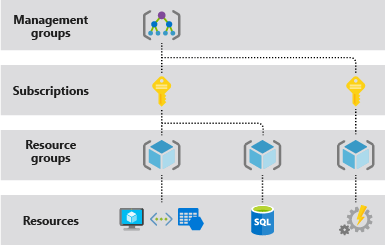
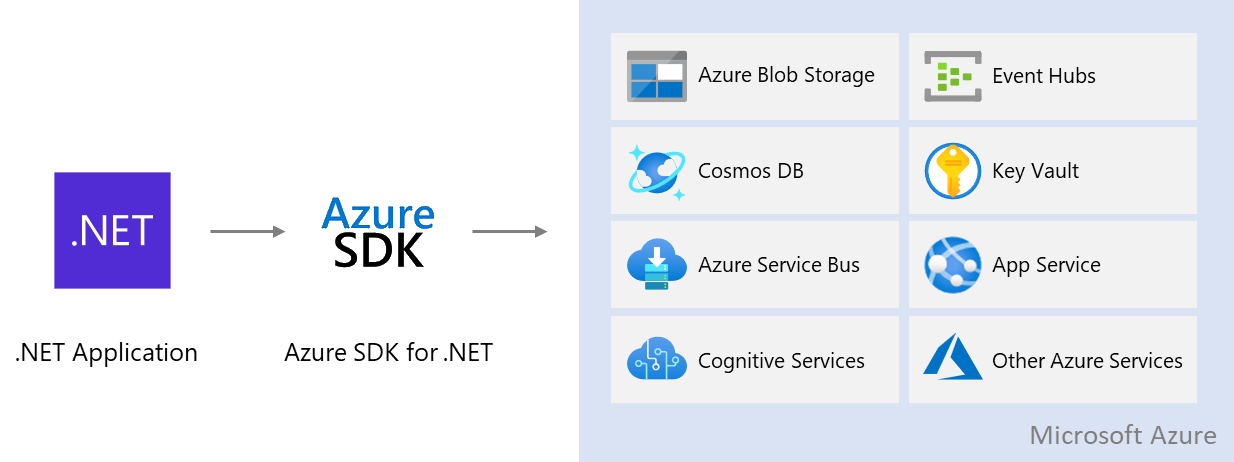
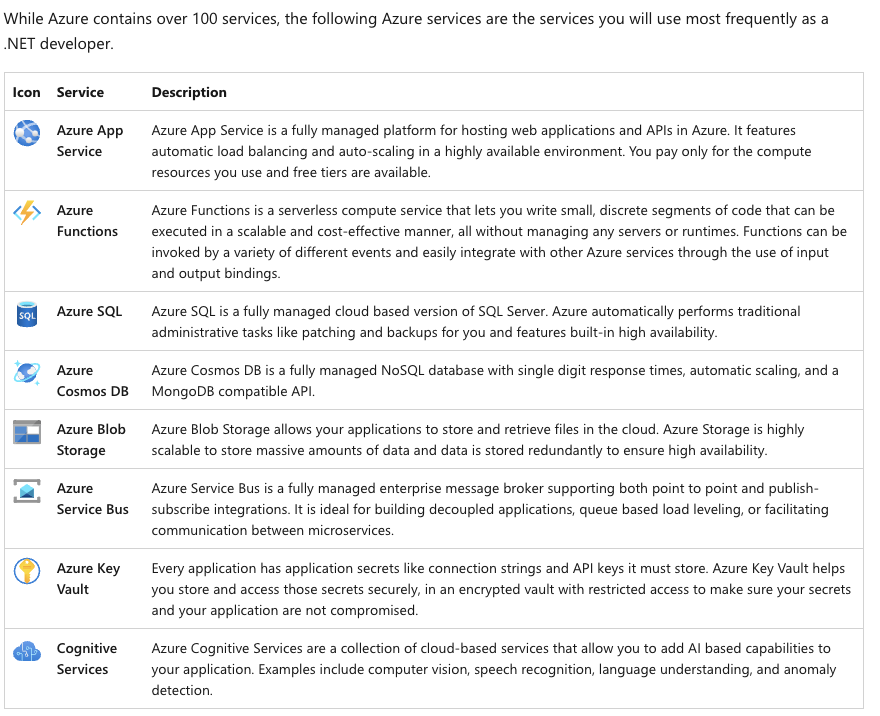
**Azure Development (Az-204)**



**Access Azure services from .NET applications**



# **Key Azure Services for .NET developers**

****

# **Deploy an ASP.NET web app**

[**https://docs.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore?tabs=net60&pivots=development-environment-vs**](https://docs.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore?tabs=net60&pivots=development-environment-vs)

# **Deploy an ASP.NET Core and Azure SQL Database app to Azure App Service**

[**https://docs.microsoft.com/en-us/azure/app-service/tutorial-dotnetcore-sqldb-app?tabs=azure-portal%2Cvisualstudio-deploy%2Cdeploy-instructions-azure-portal%2Cazure-portal-logs%2Cazure-portal-resources**](https://docs.microsoft.com/en-us/azure/app-service/tutorial-dotnetcore-sqldb-app?tabs=azure-portal%2Cvisualstudio-deploy%2Cdeploy-instructions-azure-portal%2Cazure-portal-logs%2Cazure-portal-resources)

# **Map an existing custom DNS name to Azure App Service**

[**https://docs.microsoft.com/en-us/azure/app-service/app-service-web-tutorial-custom-domain?tabs=a%2Cazurecli**](https://docs.microsoft.com/en-us/azure/app-service/app-service-web-tutorial-custom-domain?tabs=a%2Cazurecli)

# **Secure a custom DNS name with a TLS/SSL binding in Azure App Service**

[**https://docs.microsoft.com/en-us/azure/app-service/configure-ssl-bindings**](https://docs.microsoft.com/en-us/azure/app-service/configure-ssl-bindings)

# **Continuous deployment to Azure App Service**

[**https://docs.microsoft.com/en-us/azure/app-service/deploy-continuous-deployment?tabs=github**](https://docs.microsoft.com/en-us/azure/app-service/deploy-continuous-deployment?tabs=github)

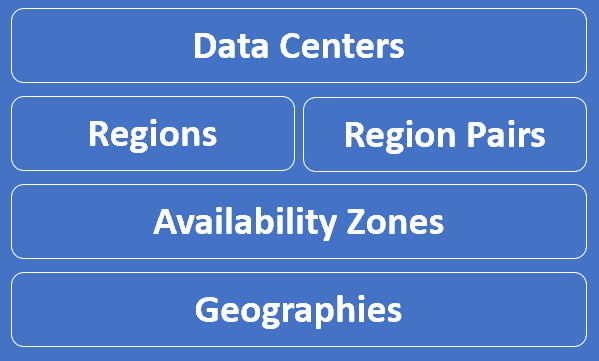
# **Host a RESTful API with CORS in Azure App Service**

**<https://docs.microsoft.com/en-us/azure/app-service/app-service-web-tutorial-rest-api>**

# **Create App Service app using an ARM template**

[**https://docs.microsoft.com/en-us/azure/app-service/quickstart-arm-template?pivots=platform-windows**](https://docs.microsoft.com/en-us/azure/app-service/quickstart-arm-template?pivots=platform-windows)

**Azure Datacenters**



An Azure data center is a unique physical building that contains thousands of physical servers with it's own power, cooling and networking infrastructure. These data ceneters are located all over the globe. As of Now there are over **160+ Azure datacenters.** These dataceneters are organised into Azure Regions.

## Why are Azure Geographies important ?

Well, regulated data like **financial**, **health care or credit data** may not be allowed to leave the country.

Legally your organisation is required to store such data in the same country where the operations are being carried out.

So, if you select a geography like India for example, Azure ensures your data is not stored outside of India.

Similarly if you select United States, the data stays inside the states.

You have complete control on which geographies you want your data and applications to be deployed.

You can be assured, Microsoft will not store customer data outside the geography you specify, except for certain non-regional services.

## What is an Azure Region ?

* An Azure Region is a set of Datacenters that are connected through a dedicated low-latency network.
* How many datacenters does a region contain. Well, we do not have a fixed number. It varies. There are regions of different sizes. A Region could be made up of just 1 dataceneter or multiple datacenters.
* The point is, an Azure Region is a group of one or more Azure Datacenters. As of this course recording, Azure has 58 regions worldwide.

## azure regions explained

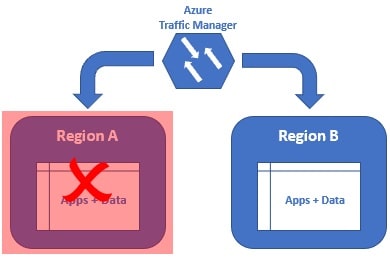
* You have the flexibility to deploy your applications and data to any Azure region you want.
* You can even deploy across multiple regions to deliver cross-region resiliency.

## What is cross-region resiliency ?

Well, in general, resilience is the ability of a software to react to problems in one of its components and still provide the best possible service.

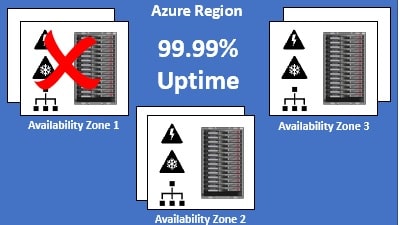
Both your software and the underlying infrastructure must be resilient. If there is a problem, the end user should not know about it. The request must be handled and processed by another region. The end user should get the same level of service.

We can get this resiliency, by deploying our application and data in at least 2 regions. In this example we have our application and data deployed in two regions - Region A and Region B.



If there is a region level failure, for example, let's say Region A has gone down. The Azure Traffic Manager is smart enough to send all the requests to Region B. The end user gets the same response. He does not even know there is a region level failure. When Region A is back online, the Azure Traffic Manager will distribute the traffic between both the regions again.

## What is an Azure Availability Zone ?

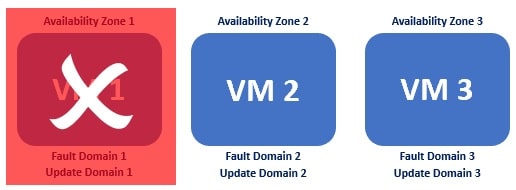
* An Azure Availability Zone is a unique physical location within an Azure region. Each Availability Zone is made up of one or more datacenters with independent power, cooling, and networking. Not all Regions have Availability Zones.
* Regions that support Availability Zones have a minimum of **three separate zones** to ensure resiliency.
* 
* If one of the Availability Zones has gone down for some reason, we still have our applications and data available from the rest of the two Availability Zones.
* There is a physical separation between each Availability Zone and it is this separation that protects our applications and data from Datacenter failures.
* With Availability Zones, Azure offers industry best 99.99% VM uptime SLA.

## Azure fault domain and update domain ?

* An **Availability Zone** is a combination of a **fault domain and an update domain**.
* For example, if you create three VMs across three zones in an Azure region, your VMs are effectively distributed across three fault domains and three update domains.

**EX :**

For some reason, if there is a power failure or power surge, equipment failure or some other fault in Availability Zone 1, only Availability Zone 1 is affected. As the availability zones are physically separted from each other, faults from one availability zone are not spread to the other availability zones. Since, in this example, we have 3 availability zones, it's like we have 3 separate fault domains. If there is a fault and Availability Zone 1 is down, we still have our VM available from the rest of the two fault domians i.e Availability Zones 2 and 3.



Similarly, if there is an update or a patch to be applied, azure schedules these at different times for different availability zones. So this means, we have just one of the availability zones affected while the update is being applied. The rest of the 2 zones are unaffected.

In our example, since we have 3 availability zones, the updates will be applied at 3 different times. So even, if the VM from Availability Zone 1 is down, due to the update being applied, we have the rest of the 2 VMs still up and running i.e the VMs in Availability Zones 2 and 3.

## Azure Zonal services and Zone-redundant services ?

If you want to make the best use of availability zones and build high-availability systems, you will have to identify your compute, storage, networking, and data resources within an availability zone and replicate them in the other availability zones. Azure services that support Availability Zones are classified into 2 categories:



**Zonal services –**

**Virtual machine** is an example of a zonal service.

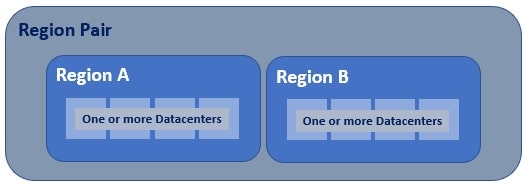
* A zonal service is pinned to a specific availability zone. This means it is only available in the availability zone where it is created.
* It is not automaically replicated to other availability zones. So, if want to build highly-available systems, identify zonal services, such as a VM for example and replicate them in other availability zones.
* Other examples of zonal services are **managed disks** and **Standard IP addresses**.

**Zone-redundant services –**

**Azure SQL Database** is an example of zone-redundant service. These zone-redundant services are automatically replicated by the Azure platform across all availability zones. Unlike Zonal services, we don't have to replicate them manually.

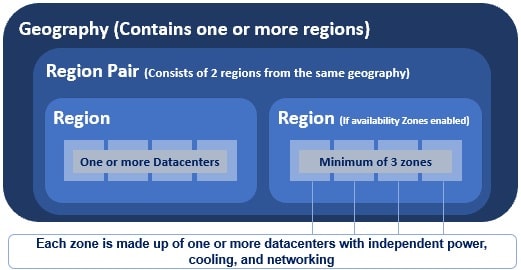
## What are Azure paired regions ?

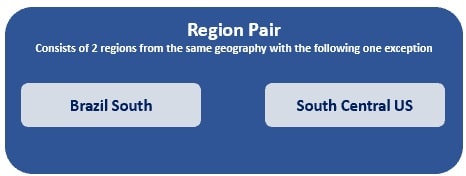
Azure regional pair, paired regions, or region pair, all these terms are used interchangeably and they refer to the same thing, i.e a pair of azure regions. So in simple terms, **a regional pair consists of two regions within the same geography.**



As you can see from the image, at the highest level we have an **Azure Geography**. As we have already discussed, an **Azure geography is an area of the world that contains one or more Azure Regions**.

For example, India, United States, Europe, Asia Pacific are a few examples of Azure Geographies.



* An azure region is made up of one or more datacenters.
* If availability zones are enabled, an azure region contains a minimum of three availability zones.
* An Availability Zone is made up of one or more datacenters. So the point is, an Azure region contains one or more datacenters or 3 or more availability zones if enabled.
* Most regions in a geography are paired to ensure business continuity and disaster recovery (BCDR). The following are a few examples of azure paired regions.
* 
* A regional pair consists of two regions within the same geography. However, there is one exception to this. Brazil South region is paired with South Central US region, which obviously, is outside of Brazil's geography.
* 

## Benefits of paired regions

**Physical separation between datacenters** : When possible, there is at least 300 miles of separation between datacenters in a regional pair, although this isn't practical or possible in all geographies. Physical datacenter separation reduces the likelihood of natural disasters, civil unrest, power outages, or physical network outages affecting both the regions at the sametime. So for whatever reason, if one of the regions is down, we still have the other region available.

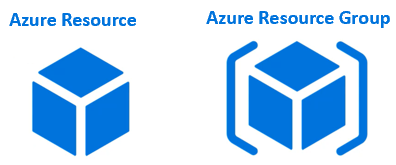
**Region recovery in the event of an outage :** If for whatever reason, several regions world-wide are down, azure prioritizes recovery of one region out of every pair. So if you want your apps and data to be highly available, deploy them in paired regions. With this setup, if both the regions are down, azure prioritizes to recover at least one region from the pair, so we have our apps and data available again soon. If applications are deployed across regions that are not paired, recovery might be delayed, in the worst case the chosen regions may be the last two to be recovered.

**Automatic Platform-provided replication :** Some services such as Geo-Redundant Storage provides automatic replication to the paired region. This is a great benefit. In an event, where one of the regions go down, you still have the data available from the other region in the region pair.

**Data residency, compliance and legal requirements :** With the exception of Brazil South, regions with in a region pair are from the same geography. This helps us meet data residency, compliance and legal requirements.

**Sequential system updates :** From time to time, patches and software updates need to be applied. Regions in a region pair are never updated simultaneously at the same time. They are always applied sequentially. This reduces the downtime to a great extent. If the planned update contains any bugs or logical errors, only one region is affected. Our apps and data will still be available from the other region in the region pair.

# **Azure Resource & resource groups**



## Azure All Services Lists :

## <https://portal.azure.com/#allservices>

Resources are instances of azure services that you create, like virtual machines, app services, storage accounts, SQL databases, function apps etc. All these are azure services. Everytime you create an instance of a service, you are creating a resource.

Anytime you create a resource, you also need to specify a resource group.

**What is an Azure Resource Group ?**

As the name implies, a Resource Group is a group of azure resources like virtual machines, app services, storage accounts, SQL databases etc. It's a logical container for grouping related azure resources.

## Azure Resource Group Example

Let's say we are developing a web application. There are several ways to do this. To keep this example simple, let's just assume we need the following 3 azure resources.

1. Virtual Machine - To host and run our web application
2. Storage Account - To store images, videos and other resources that our web application needs
3. SQL database - To store our application data

Let's say for this example sake we have the following environments. Most organisations have these deployment environments.

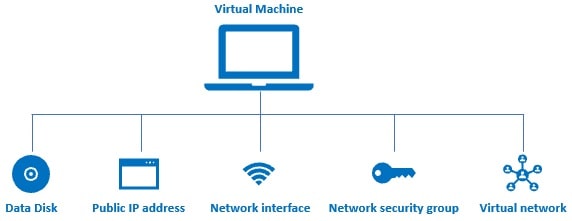
1. Development
2. Testing
3. Staging
4. PreProduction
5. Production

Let's say our web application name is morningAI.com. We might create the following 4 resource groups, one for each environment.

1. rg-morningAI-development
2. rg- morningAI -staging
3. rg- morningAI -preproduction
4. rg- morningAI -production

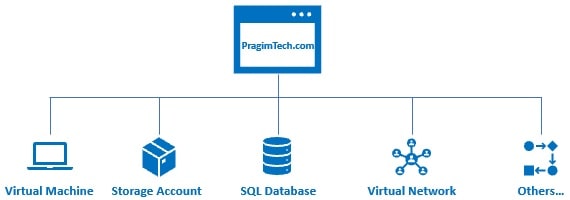
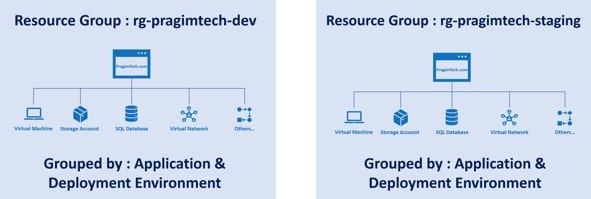
**rg-<applicationName>-<deploymentEnvironment>**

## Azure resource groups - Imporantant points to remember

1. An azure resource is any service instance that you create. For example, virtual machine, Azure sql database, storage account etc.
2. A resource group, as the name implies, a group of related azure resources.
3. In general, resources in a resource group share the same life cycle, so they can be easily created, deployed, updated, and deleted as a single unit.
4. When you create a resource group, you specify a region. It is this region where the meta-data about the resource group is stored.
5. However, the resources themselves can be in any azure region.
6. Resources in one resource group can interact with resources in other resource groups.
7. Each resource must be in one and only one resource group. You cannot have a resource in more than 1 resource group at the sametime.
8. You can move a resource from one resource group to another.
9. You can add or remove a resource from a resource group at any time.
10. You can group resources any way you want. Anyway that makes sense to your oragnisation really - By department, By country, By application, By resource type or a combination of these.
11. 

When we create a virtual machine in azure several other associated resources like the following are created.

1. a data disk for the virtual machine
2. Public IP address
3. Network interface
4. Network security group
5. Virtual network

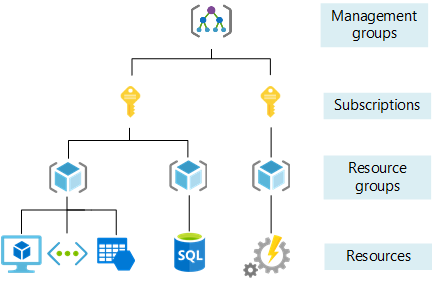
* Without these resources, an azure virtual machine doesn't work as expected. After you are done with the VM, you may want to delete it to save on cost. However, when you delete the VM the associated resources are not automatically deleted. You have to delete them manually. If you forget to delete 1 or more associated resources, you are unnecessarily paying for those resources that you are not actually using.
* On the other hand, if you create a virtual machine in a resource group, all the other associated resources are also created in the same resource group. When we delete the resource group, not just the virtual machine, all it's associated resources are also automatically deleted.
* Another example. Let's say, you are creating a web application. To aid this you need several resources like a virtual machine,  storage account, sql database, virtual networks and many other dependant and related services.
* 
* Without resource groups, if you have to develop and deploy this application, you have to manually create all these azure resources. That too, you have to create them in the right order. If it's just one time, then it's okay. But in real-world, with every company using agile approach and CI/CD i.e Continuous Integration & Continuous Deployment, applications are deployed several times a day. For example, everytime, a new piece of code is checked-in to the source control, a new build is deployed to the test-environment. So everytime we have to do this, if we have to create all the resources manually, it's not only tedious and time consuimg, but also error-prone. What if you create the resources in the wrong order or even worse what if you forget to create a resource. It gets even more messy and complicated if you have to manage multiple applications and multiple environments.
* With resource groups, you can group related resources any way you want, may be for example, by application and by deployment environment. This grouping obviously allows you to manage all the resources that belong to a specific application and deployment environment as one unit. You can even automate deployments using Azure Resource Manager Templates.
* 

## Role-based access control (RBAC)

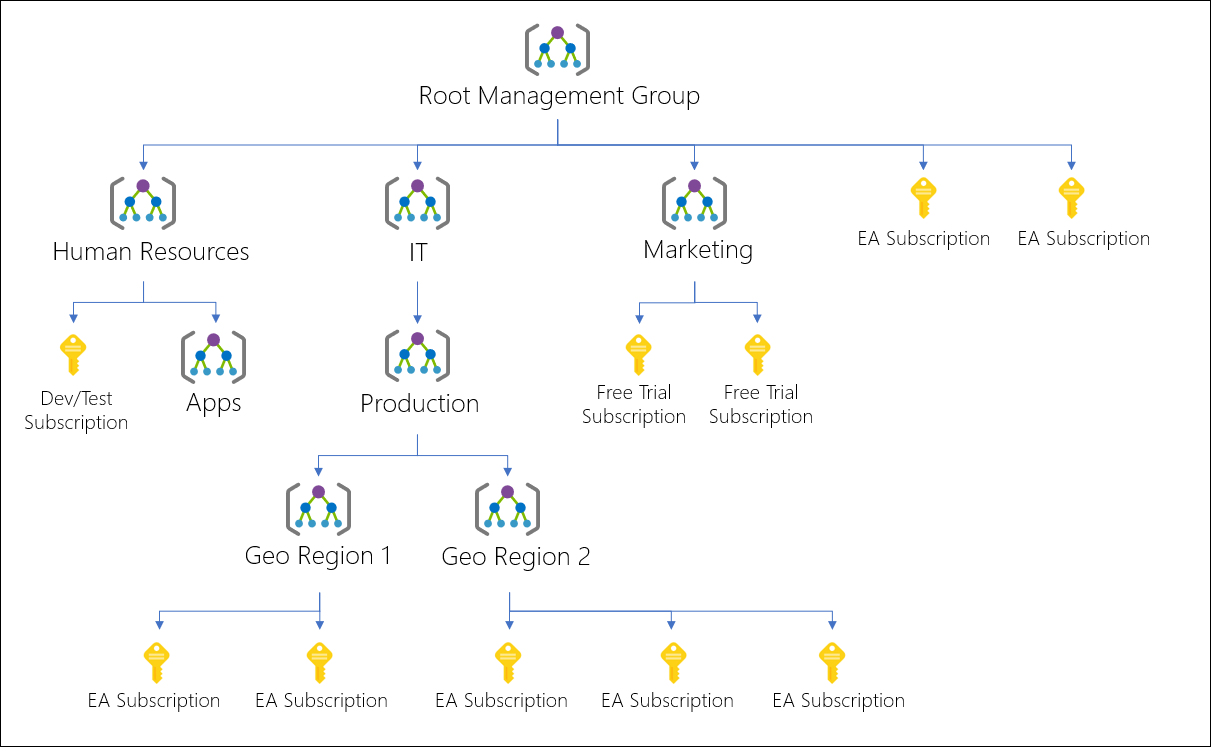
Role-based access control (RBAC) can be applied at the resource group level. This makes it much easier to manage user access to the resources in the group. When the users log into the azure portal, they will only see resource groups they have access to and not others within the subscription. Administrators will still be able to assign access control for users to individual resources within the resource group based on their roles.

# **Azure management groups and subscriptions ?**

* If there are only a few subscriptions in your organisation, then it's relatively simple to manage them independently. However, in an organisation there are usually many employees and may be, many applications. If all these employees are provided azure subscriptions and if they start creating azure resources at will, it may soon become difficult to control, manage and track who is creating what and eventually the costs may go out of control.
* So, Azure has four levels of management-scope to organize, secure, manage and track the costs. The following image from MSDN shows the four levels of management-scope and the relationship between them.



### **Management groups**

* Management group is at the top of the hierarchy. All subscriptions in a management group automatically inherit the conditions or settings specified at the management group level.
* So, a management group is like a container for all your subscriptions. Just like how there can be multiple subscriptions, there can also be multiple management groups in an organisation.
* 
* For whatever reason, let's say, in our organisation, we want to allow azure resources to be created only in the East US region. One easy way to do this is to create such a policy at the IT Management Group level. This policy is then automatically enforced on all the Management Groups and Subscriptions that are descendants of the IT management group. The descendants will not be able to alter this security policy in any way and it is also applicable to all resources under those subscriptions. So, obviously governance becomes much easier.
* Management settings like policies and role-based access control can be applied at any of the management levels. The level you select determines how widely the setting is applied. Lower levels inherit settings from higher levels. For example, when you apply a policy to a subscription, that policy is also applied to all resource groups and resources in that subscription. In general, it makes sense to apply critical settings at higher levels and project-specific settings at lower levels.

### **Subscriptions**

A subscription sits under a management group. It associates user accounts and the resources that were created by those user accounts. Each subscription has limits or quotas on the amount of resources you can create and use. Organizations can use subscriptions to manage costs and the resources that are created by users, teams, or projects.

### **Resource groups**

A resource group, as the name implies, is a group of related azure resources. It is basically a logical container into which Azure resources like web apps, databases, and storage accounts are deployed and managed. We discussed resource groups in detail in Parts 5 and 6 of this azure tutorial.

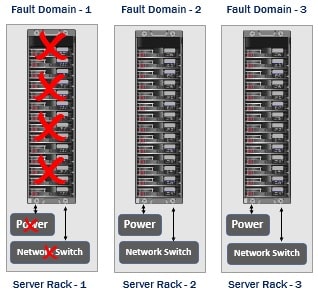
### **Resources**

An azure resource is any service instance that you create. For example, virtual machine, Azure sql database, storage account etc.

# **Azure availability set**

## Azure Fault Domain

What does an azure dataceneter contain? Well, in simple terms, it contains several racks of servers. Each rack in turn, may contain several servers with it's own power supply and network switch. In reality, server racks are much complicated than this. They may be equipped with redundant power supplies and network switches. However, to keep this example simple, let's just say, a rack contains 15 to 20 physical servers with it's own power supply and network switch.

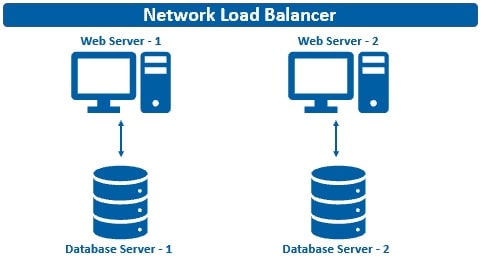


## Azure Update Domain

The server hardware and supporting infrastructure in a datacenter is divided in to multiple fault domains and update domains. An update domain is a group of resources that can be updated and rebooted if required at the same time. From time to time, patches and software updates need to be applied. Some updates require servers to be rebooted. Only one update domain is rebooted at a time. A rebooted update domain is then given 30 minutes to recover before maintenance is initiated on a different update domain. This reduces the downtime to a great extent.

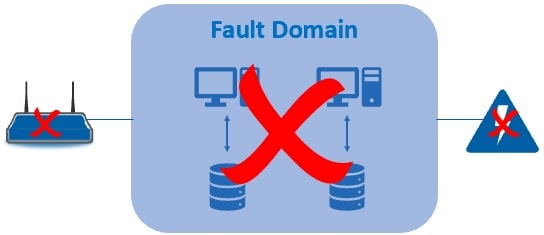
* So, if you want your azure resources, like virtual machines for example to be available even during the update process, have them deployed across multiple update domians.

## Azure Availability Set

* An Availability Set is a logical grouping for isolating virtual machine resources from each other. Azure makes sure that the VMs we place in an Availability Set run across multiple physical servers, compute racks, storage units, and network switches. If a hardware or software failure happens, only a subset of our VMs are impacted and our overall solution still stays operational. Availability Sets are essential for building reliable cloud solutions.
* Let's say we have a simple two tier web application. On one of the virtual machines, we have a web server and on another virtual machine, we have our database server. Now, to be able to handle and process more requests we have 2 web servers and 2 database servers. In real-world, web applications that have lot of demand, for example Google.com, Gmail.com, and Amazon.com may have many many web servers and database servers. However, to keep our example simple, let's just stick to two web servers and two database servers. The load balancer obviously distributes the incoming traffic between the two web servers.
* 

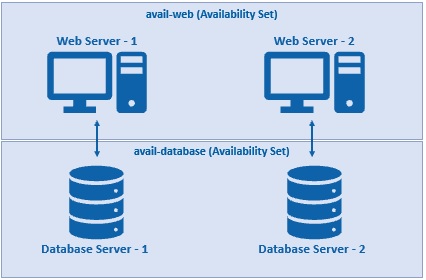
## What may happen if availability sets are not used ?

Well, all the 4 VMs (i.e the two web servers and two database servers) may end up in the same fault domain or update domain. As a result if there is a software failure or hardware failure like power supply or network switch failure, all your web and databaser servers go down. End result your web application is no longer available. If it's an e-commerce application like amazon.com for example, just imagine the extent of loss to the business every second the system is down.

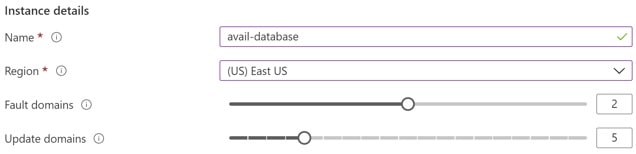


## Use availability sets for high availability

Since we have two tiers - a web tier and a database tier, we create 2 availability sets - one for the web tier and the other for database tier.



In Azure, when creating an availability set, we specify the following :



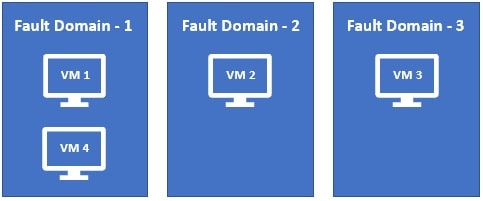
### **Name**

The name of the of the availaility set. It's a common convention to use the prefix "avail" for availability sets

### **Region**

Azure region where we want the resources to be deployed

### **Fault domains**

* The number of fault domains you want in the availability set. For example, if you set the fault domains to 3 and you create 3 virtual machines, each of them will be placed in 3 separate fault domains. If there is a fault like a power failuer for example, only one of the server racks is affected. This means only one of you VM is down, but the other 2 vms from the other 2 fault domains are still available. This in turn means, your workload i.e in this case your web application is still available to end users.
* What happens if we create a fourth VM with 3 fault domains. Well, it will be placed in one of the 3 fault domains. This means, in one of the 3 fault domains, you will have 2 VMs and the rest 2 will have 1 each.
* 

## Availability Sets and Virtual Machine SLA

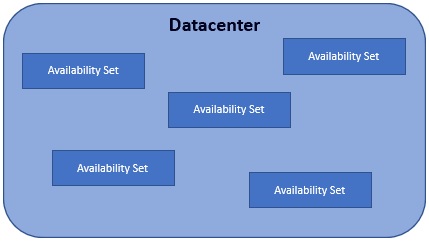
For all Virtual Machines that have two or more instances deployed in the same Availability Set, Microsoft guarantees, you will have Virtual Machine Connectivity to at least one instance at least 99.95% of the time.

**You can't add an existing Virtual Machine to an availability set after it's created.** So, if you want a virtual machine in an availability set, it's a decision you have to make at the time of creation, not after it is created.

# **Azure availability set vs availability zone**

## Update Domains and Fault Domains in an Availability Set

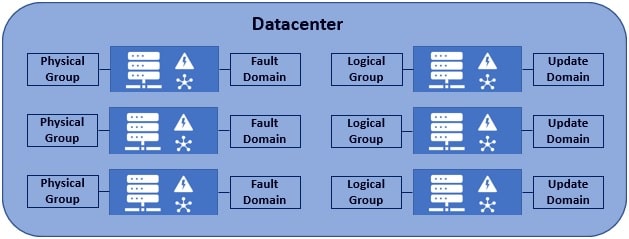
Availability set is a concept with in a datacenter.



The servers in a datacenter are divided into multiple physical and logical groups.

* The physical grouping is called **fault domain** and the logical

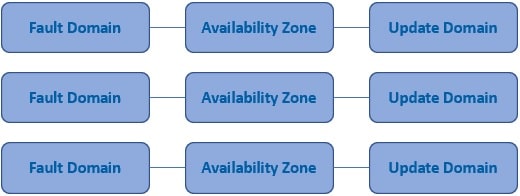
grouping is called **update domain**.

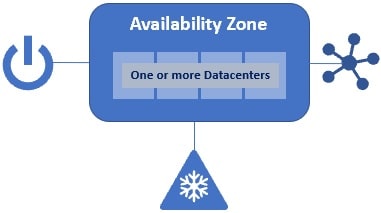
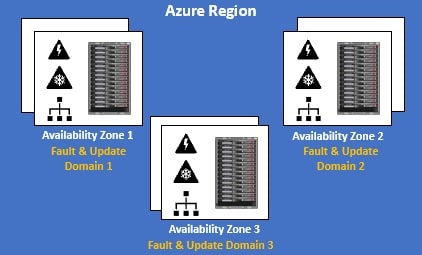
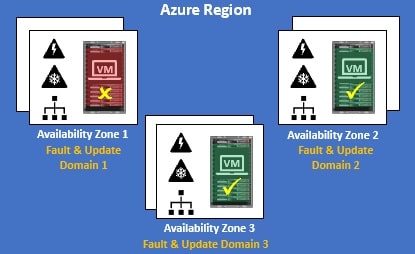


So, by grouping servers in a datacenter into multiple racks i.e multiple fault domains, we are able to eliminate a single point of failure.

* an update domain is a logical grouping. Why are update domains required? Well, from time to time, server patches and software updates need to be applied. Some updates require servers to be rebooted. Now, we don't want all the servers to be rebooted at the sametime. Only one update domain is rebooted at a time. A rebooted update domain is then given 30 minutes to recover before maintenance is initiated on a different update domain. So, in short, an update domain is a group of servers that can be updated and rebooted at the same time.
* An availability set is a concept with in a datacenter and it is made up of multiple fault domains and update domains. In fact, when you create an availability set, you can specify how many fault domains and update domains you want in that availability set.
* 

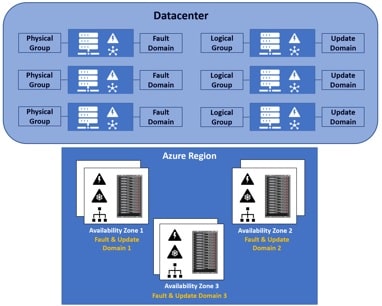
## Update Domains and Fault Domains in an Availability Zone

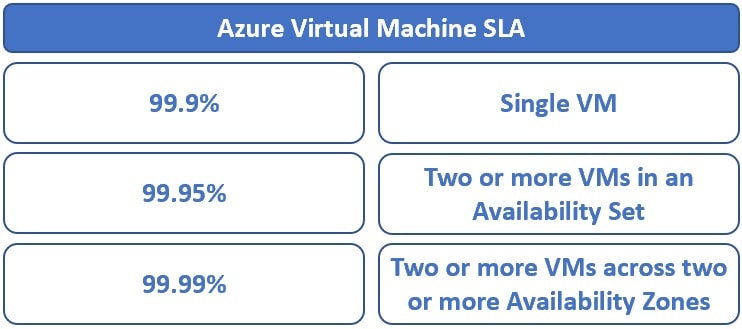


* Well, an Azure Availability Zone is a unique physical location within an Azure region.
* Each Availability Zone is made up of one or more datacenters with independent power, cooling, and networking.
* 
* Not all Regions have Availability Zones, but regions that do have availability zones, have a minimum of three separate zones.
* 
* You can think of each availability zone as a separate fault domain and update domain. So in a given azure region if you have 3 availability zones, then it's like you have 3 fault domains and 3 update domains.
* 
* If one of the Availability Zones has gone down for some reason, we still have 2 VMs from the rest of the 2 availability zones. Similarly, if there is an update or a patch to be applied, azure schedules these at different times for different availability zones. So this means, we have just one of the availability zones affected while the update is being applied. The rest of the 2 zones are unaffected.

**NOTES**

Now, here is the important bit to keep in mind. The concept of update domain and fault domain is present in both availability set and availability zone. In case of an availability set, a group of servers (physical or logical), physical grouping i.e a rack of servers is a fault domain and logical group of servers is an update domain. In case of an availability zone, each availability zone itself is considered, a separate falut domain and update domain. So in a given azure region if you have 3 availability zones, then it's like you have 3 fault domains and 3 update domains.





## Difference between Azure Availability Set and Availability Zone

🡪 An **Availability set** protects your azure resources from failures within data centres

🡪 An **Availability zone** protects from entire datacenter failures.

From the Service Level Agreement (SLA) standpoint, i.e uptime and connectivity - With Availability Set azure offers 99.95% SLA where as with Availability Zone we have 99.99% SLA. So, there we go, with availability zones, Azure offers industry best 99.99% VM uptime SLA.

