Contents

[CLOUD COMPUTING 3](#_Toc109037214)

[IP ADDESSS 3](#_Toc109037215)

[CLOUD CONCEPTS 5](#_Toc109037216)

[WHAT IS CLOUD COMPUTING 5](#_Toc109037217)

[VIRTUALIZATION 5](#_Toc109037218)

[BENEFITS OF VIRTUALIZATION 6](#_Toc109037219)

[TYPES OF CLOUD MODELS 6](#_Toc109037220)

[BENEFITTS OF CLOUD 7](#_Toc109037221)

[CAPITAL EXPENSES VS. OPERATING EXPENSES 7](#_Toc109037222)

[CLOUD COMPUTING IS A CONSUMPTION-BASED MODEL 7](#_Toc109037223)

[WHAT ARE CLOUD SERVICE MODELS? 7](#_Toc109037224)

[IAAS 8](#_Toc109037225)

[PAAS 8](#_Toc109037226)

[SAAS 9](#_Toc109037227)

[CLOUD SERVICE MODEL COMPARISON 9](#_Toc109037228)

[WHAT IS SERVERLESS COMPUTING? 9](#_Toc109037229)

[AZURE 10](#_Toc109037230)

[HOW AZURE WORKS? 10](#_Toc109037231)

[AZURE MARKET PLACE 10](#_Toc109037232)

[AZURE SUBSCRIPTIONS, MANAGEMENT GROUPS, AND RESOURCES 11](#_Toc109037233)

[WORKLOAD 11](#_Toc109037234)

[RESOURCE GROUP 11](#_Toc109037235)

[CREATING A RESOURCE GROUP 12](#_Toc109037236)

[AZURE RESOURCE MANAGER 12](#_Toc109037237)

[THE BENEFITS OF USING RESOURCE MANAGER 13](#_Toc109037238)

[REGION AND ZONES 13](#_Toc109037239)

[WHY REGION IS IMPORTANT 13](#_Toc109037240)

[SELECTING A REGION 14](#_Toc109037241)

[AZURE AVAILABILITY ZONE 14](#_Toc109037242)

[AZURE REGION PAIR 14](#_Toc109037243)

[AZURE CORE SERVICES 15](#_Toc109037244)

[AZURE COMPUTE SERVICES 16](#_Toc109037245)

[VIRTUAL MACHINE 16](#_Toc109037246)

[AZURE NETWORK SERVICES 34](#_Toc109037247)

[AZURE VIRTUAL NETWORK 34](#_Toc109037248)

[NETWORK SECURITY GROUP 38](#_Toc109037249)

[NETWORK CONNECTIVITY OPTIONS 39](#_Toc109037250)

[AZURE STORAGE SERVICE 41](#_Toc109037251)

[CREATING A STORAGE ACCOUNT (GENERAL PURPOSE V2 STORAGE ACCOUNT) 42](#_Toc109037252)

[STORAGE ACCOUNT 42](#_Toc109037253)

[TYPES OF STORAGE ACCOUNT 43](#_Toc109037254)

[DATA REDUNDANCY IN AZURE STORAGE ACCOUNT 43](#_Toc109037255)

[CORE STORAGE SERVICES 46](#_Toc109037256)

[AZURE COSMOS DATABASE 54](#_Toc109037257)

[CREATING A COSMOS DB ACCOUNT 55](#_Toc109037258)

[AZURE SQL DATABASE 59](#_Toc109037259)

[AZURE WEB-APP SERVICE 61](#_Toc109037260)

[ADVANTAGE OF USING APP SERVICE 62](#_Toc109037261)

[CREATING AN APP SERVICE 62](#_Toc109037262)

[AZURE APP SERVICE PLAN 63](#_Toc109037263)

[AZURE LOAD BALANCER 64](#_Toc109037264)

[COMPONENTS OF LOAD BALANCER 65](#_Toc109037265)

[BASIC LOAD BALANCER 66](#_Toc109037266)

[STANDARD LOAD BALANCER 66](#_Toc109037267)

# CLOUD COMPUTING

## IP ADDESSS

* IP is the unique identifier of a device in a network.



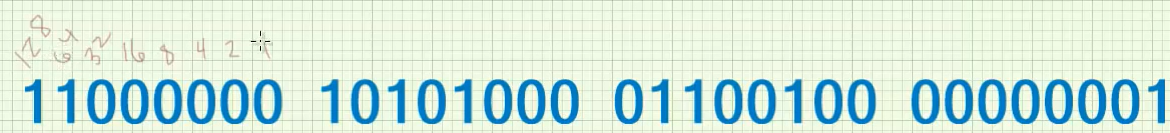
|  |  |
| --- | --- |
|  | * The IP address has 2 parts – The Network part and a Host part. * Each host in a network will have same Network Address |

#### IPv4 ADDRESSING

* It’s a 32-bit logical address
* It consists of 4 octet – and each octet ranges from 0 -255
* IP address has parts ***Network ID and Host ID***



#### BINARY TO DECIMAL



* When the IP is represented in Binary the decimal representation can be done using the ***power to 2***
* Hence the equivalent decimal will be 192.168.100.1 (128+64).(128 + 32 +8).(64+32+4).(1)

#### CLASSES IN IP ADDRESSING



* Ranges 127.x.x.x are reserved for the [loopback or localhost](https://www.computerhope.com/jargon/l/locahost.htm), for example, 127.0.0.1 is the loopback address.
* Range 255.255.255.255 [broadcasts](https://www.computerhope.com/jargon/b/broadcas.htm) to all hosts on the local network.

1. HOW TO DECIDE THE CLASS OF IP ADDRESS?

To decide the class of IP address we consider the first octet for example – **132**.20.10.192 – This IP belong to CLASS B.

#### NETWORK ID IN IP ADDRESSING

|  |  |
| --- | --- |
|  | * In Class A IP address 1st 8 bits are Network Id * Remaining 24 bits are for Host hence for class a we can have * Hence number of Host is **224 = 16777216** |
| * In Class B IP address 1st IP 16 bits are Network Id * Remaining 16 bits are for Host hence for class a we can have * Hence number of Host is **216 = 65536** |
| * In Class C IP address 1st IP 24 bits are Network Id * Remaining 8 bits are for Host hence for class a we can have * Hence number of Host is **216 = 256** |

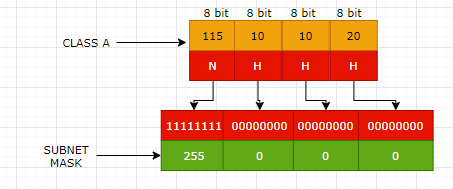
* Example : Find the network id of IP : **192.10.10.10** ?
  + Step 1 : The class of this IP is Class C
  + Step 2: The class B IP has first 24 bit (1st 3 Octet) as its network id . Hence the Network id : **192.10.10.0**

#### SUBNET, SUBNET MASK AND CIDR NOTATION

* Note the network id is represent by “1” and host is represented by “2”.

##### CALCULATING SUBNET MASK

* Example : Calculate the Subnet mask of : **115.10.10.20 ? Ans – 255.0.0.0**



###### WHAT IS THE JOB OF SUBNET MASK?

* Since the IP is consist of 2 part Network and Host part. Network Part in a network will be same for all host.
* Host decide the network Id with the help of Subnet Mask.
* Subnet mask is also of 32 bits -which has a mapping with the IP address. The 1s represent the network portion and 0s are the host portion. In the above example – When an IP is given to a device – then the subnet mask is also configured. The bits represented with “1” is the network id in the IP – when compared from left to right.

##### BROADCAST ID

* Broadcast IP is used to broadcast to all the host in the network.
* Find the class, network id ,broadcast id and usable IP of the following IP

|  |  |
| --- | --- |
| 150.10.20.30 | * The IP belongs to Class B * Network Id: 150.10.0.0 * Broadcast ID: To calculate the broadcast id, set the host part of Network to 255. Hence the broadcast Id - 150.10.255.255 * Usable Host IP = Total Number of IP address – (Network IP + Broadcast IP) i.e.   + For a network there will be on Network IP and one Broadcast IP.   + Usable Host IP = Total IP -2= 216 – 2 = 65536 – 2= **65534** |

###### CIDR NOTATION (CLASSLESS INTER DOMAIN ROUTING)

* In class world we use subnet mask to deduce the number of bits used for Network
* Similarly, rather than using the subnet mask - CIDR notation is a way to represent the network.
* For example, **192.168.100.1/24**
  + This represent that the first 24 bits are used for network id. It is also called network pre-fix

# CLOUD CONCEPTS

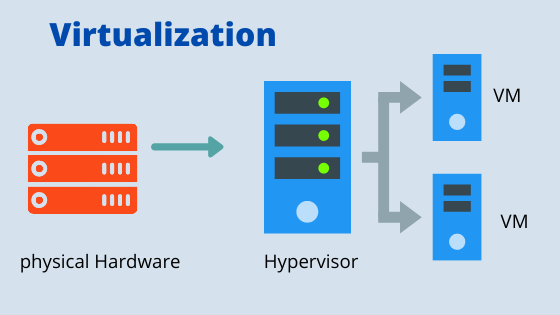
## WHAT IS CLOUD COMPUTING

1. **It's the delivery of computing services over the internet, which is otherwise known as the cloud.** These services include **servers, storage, databases, networking, software, analytics, and intelligence.** Cloud computing offers faster innovation, flexible resources, and economies of scale.
2. Cloud computing is the delivery of computing services over the internet by using a pay-as-you-go pricing model. You typically pay only for the cloud services you use, which helps you:

* Lower your operating costs.
* Run your infrastructure more efficiently.
* Scale as your business needs change.

To put it another way, cloud computing is a way to rent compute power and storage from someone else's datacenter. You can treat cloud resources like you would resources in your own datacenter. When you're done using them, you give them back. You're billed only for what you use. Instead of maintaining CPUs and storage in your datacenter, you rent them for the time that you need them. The cloud provider takes care of maintaining the underlying infrastructure for you.

## VIRTUALIZATION



* Virtualization is the process of running a virtual instance of a computer system in a layer abstracted from the actual hardware. Most commonly, it refers to running multiple operating systems on a computer system simultaneously.
* To the applications running on top of the virtualized machine, it can appear as if they are on their own dedicated machine, where the operating system, libraries, and other programs are unique to the guest virtualized system and unconnected to the host operating system which sits below it.

Note: The machine on which VM are created are called HOST MACHINE and all the VM are called GUEST MACHINE.

* For administrators of servers, virtualization also offers the ability to run different operating systems, but perhaps, more importantly, it offers a way to segment a large system into many smaller parts, allowing the server to be used more efficiently by several different users or applications with different needs. It also allows for isolation, keeping programs running inside of a virtual machine safe from the processes taking place in another virtual machine on the same host.

### BENEFITS OF VIRTUALIZATION

* Better resource utilization
* Lowers the cost of resources
* Enable Remote Access
* Pay as you go
* Enable running multiple OS

## TYPES OF CLOUD MODELS

|  |  |
| --- | --- |
| CLOUD MODELS | DESCRIPTION |
| **PUBLIC CLOUD** | Services are offered over the public internet and available to anyone who wants to purchase them. Cloud resources, such as servers and storage, are owned and operated by a third-party cloud service provider and delivered over the internet.   * No capital expenditures to scale up. * Applications can be quickly provisioned and deprovisioned. * Organizations pay only for what they use. |
| **PRIVATE CLOUD** | A private cloud consists of computing resources used exclusively by users from one business or organization. A private cloud can be physically located at your organization's on-site (on-premises) datacenter, or it can be hosted by a third-party service provider.   * Hardware must be purchased for start-up and maintenance. * Organizations have complete control over resources and security. * Organizations are responsible for hardware maintenance and updates. |
| **HYBRID CLOUD** | A hybrid cloud is a computing environment that combines a public cloud and a private cloud by **allowing data and applications to be shared between them**.   * Provides the most flexibility. * Organizations determine where to run their applications. * Organizations control security, compliance, or legal requirements. |

## BENEFITTS OF CLOUD

* **HIGH AVAILABILITY**: Depending on the service-level agreement (SLA) that you choose, your cloud-based apps can provide a continuous user experience with no apparent downtime, even when things go wrong.
* **SCALABILITY**: Apps in the cloud can scale vertically and horizontally:
  + Scale vertically to increase compute capacity by adding RAM or CPUs to a virtual machine.
  + Scaling horizontally increases compute capacity by adding instances of resources, such as adding VMs to the configuration.
* **ELASTICITY**: You can configure cloud-based apps to take advantage of **autoscaling**, so your apps always have the resources they need.
* **AGILITY**: Deploy and configure cloud-based resources quickly as your app requirements change.
* **GEO-DISTRIBUTION**: You can deploy apps and data to regional datacenters around the globe, thereby ensuring that your customers always have the best performance in their region.
* **DISASTER RECOVERY**: By taking advantage of cloud-based backup services, data replication, and geo-distribution, you can deploy your apps with the confidence that comes from knowing that your data is safe in the event of disaster.

## CAPITAL EXPENSES VS. OPERATING EXPENSES

There are two different types of expenses that we should consider:

* **CAPITAL EXPENDITURE (CAPEX)** is the up-front spending of money on physical infrastructure, and then deducting that up-front expense over time. The up-front cost from CapEx has a value that reduces over time.
* **OPERATIONAL EXPENDITURE (OPEX)** is spending money on services or products now, and being billed for them now. You can deduct this expense in the same year you spend it. There is no up-front cost, as you pay for a service or product as you use it.

*To summarize, CapEx requires significant up-front financial costs, as well as ongoing maintenance and support expenditures. By contrast, OpEx is a consumption-based model, so we are only responsible for the cost of the computing resources that we use.*

## CLOUD COMPUTING IS A CONSUMPTION-BASED MODEL

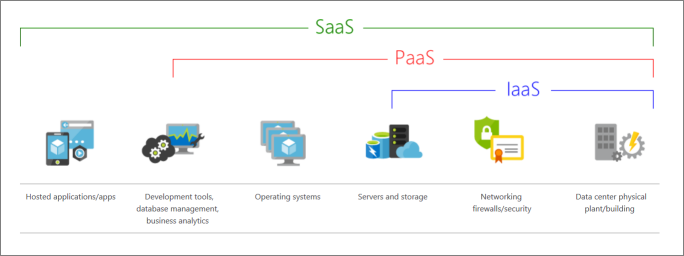
Cloud service providers operate on a **consumption-based model**, which means that end users only pay for the resources that they use. Whatever they use is what they pay for.

A consumption-based model has many benefits, including:

* No upfront costs.
* No need to purchase and manage costly infrastructure that users might not use to its fullest.
* The ability to pay for additional resources when they are needed.
* The ability to stop paying for resources that are no longer needed.

## WHAT ARE CLOUD SERVICE MODELS?

These models define the different levels of shared responsibility that a cloud provider and cloud tenant are responsible for.



|  |  |
| --- | --- |
| MODEL | DESCRIPTION |
| **Iaas**  (Infrastructure-as-a-Service) | This cloud service model is the closest to managing physical servers; **a cloud provider will keep the hardware up to date, but operating system maintenance and network configuration is up to you as the cloud tenant.** For example, **Azure virtual machines** are fully operational virtual compute devices running in Microsoft datacenters. An advantage of this cloud service model is rapid deployment of new compute devices. Setting up a new virtual machine is considerably faster than procuring, installing, and configuring a physical server. |
| **PaaS**  (Platform-as-a-Service) | This cloud service model is a managed hosting environment. The cloud provider manages the virtual machines and networking resources, and the cloud tenant deploys their applications into the managed hosting environment. For example, Azure App Services provides a managed hosting environment where developers can upload their web applications, without having to worry about the physical hardware and software requirements. |
| **SaaS**  **(**Software-as-a-Service**)** | In this cloud service model, the cloud provider manages all aspects of the application environment, such as virtual machines, networking resources, data storage, and applications. The cloud tenant only needs to provide their data to the application managed by the cloud provider. For example, Microsoft Office 365 provides a fully working version of Microsoft Office that runs in the cloud. All you need to do is create your content, and Office 365 takes care of everything else. |

### IAAS

IaaS is the most flexible category of cloud services. It aims to give you complete control over the hardware that runs your application. Instead of buying hardware, with IaaS, you rent it.

#### ADVANTAGES

* **No CapEx** -Users have no up-front costs.
* **Agility** - Applications can be made accessible quickly, and deprovisioned whenever needed.
* **Management**. The shared responsibility model applies; the user manages and maintains the services they have provisioned, and the cloud provider manages and maintains the cloud infrastructure.
* **Consumption-based model**. Organizations pay only for what they use and operate under an Operational Expenditure (OpEx) model.
* **Skills**. No deep technical skills are required to deploy, use, and gain the benefits of a public cloud. Organizations can use the skills and expertise of the cloud provider to ensure workloads are secure, safe, and highly available.
* **Cloud benefits**. Organizations can use the skills and expertise of the cloud provider to ensure workloads are made secure and highly available.
* **Flexibility**. IaaS is the most flexible cloud service because you have control to configure and manage the hardware running your application.

### PAAS

PaaS provides the same benefits and considerations as IaaS, but there are some additional benefits to be aware of.

#### ADVANTAGES

* **No CapEx**. Users have no up-front costs.
* **Agility**. PaaS is more agile than IaaS, and users don't need to configure servers for running applications.
* **Consumption-based model**. Users pay only for what they use, and operate under an OpEx model.
* **Skills**. No deep technical skills are required to deploy, use, and gain the benefits of PaaS.
* **Cloud benefits**. Users can take advantage of the skills and expertise of the cloud provider to ensure that their workloads are made secure and highly available. In addition, users can gain access to more cutting-edge development tools. They can then apply these tools across an application's lifecycle.
* **Productivity**. Users can focus on application development only, because the cloud provider handles all platform management. Working with distributed teams as services is easier because the platform is accessed over the internet. You can make the platform available globally more easily.

#### DISADVANTAGE

**Platform limitations**. There can be some limitations to a cloud platform that might affect how an application runs. When you're evaluating which PaaS platform is best suited for a workload, be sure to consider any limitations in this area.

### SAAS

SaaS is software that's centrally hosted and managed for you and your users or customers. Usually one version of the application is used for all customers, and it's licensed through a monthly or annual subscription.

SaaS provides the same benefits as IaaS, but again there are some additional benefits to be aware of too.

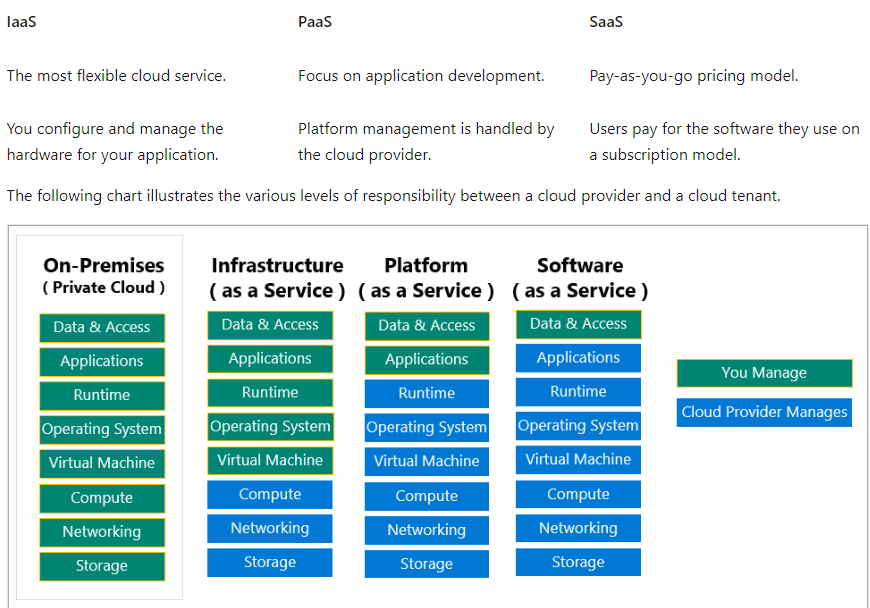
#### ADVANTAGES

* **No CapEx**. Users have no up-front costs.
* **Agility**. Users can provide staff with access to the latest software quickly and easily.
* **Pay-as-you-go pricing model**. Users pay for the software they use on a subscription model, typically monthly or yearly, regardless of how much they use the software.
* **Skills**. No deep technical skills are required to deploy, use, and gain the benefits of SaaS.
* **Flexibility**. Users can access the same application data from anywhere.

#### DISADVANTAGE

**Software limitations**. There can be some limitations to a software application that might affect how users work. Because you're using as-is software, you don't have direct control of features. When you're evaluating which SaaS platform is best suited for a workload, be sure to consider any business needs and software limitations.

### CLOUD SERVICE MODEL COMPARISON



### WHAT IS SERVERLESS COMPUTING?

* Like PaaS, serverless computing enables developers to build applications faster by eliminating the need for them to manage infrastructure.
* With serverless applications, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code.
* Serverless architectures are highly scalable and event-driven, only using resources when a specific function or trigger occurs.
* It's important to note that servers are still running the code. The "serverless" name comes from the fact that the tasks associated with infrastructure provisioning and management are invisible to the developer. This approach enables developers to increase their focus on the business logic and deliver more value to the core of the business.
* Serverless computing helps teams increase their productivity and bring products to market faster, and it allows organizations to better optimize resources and stay focused on innovation.
* Function App are example of serverless model

# AZURE

## HOW AZURE WORKS?

* Azure uses a technology known as **virtualization**. ***Virtualization separates the tight coupling between a computer's hardware and its operating system, using an abstraction layer called a hypervisor***.
* The hypervisor emulates all the functions of a real computer and its CPU in a virtual machine, optimizing the capacity of the obstructed hardware. It can run multiple virtual machines at the same time, and each virtual machine can run any compatible operating system, such as Windows or Linux.
* Azure takes this virtualization technology and repeats it on a massive scale in Microsoft data centers throughout the world. Each data center has mini racks filled with servers, and each server includes a hypervisor to run multiple virtual machines.
* A network switch provides connectivity to all those servers. One server in each rack runs a special piece of software called a fabric controller. Each fabric controller is connected to another special piece of software known as the Orchestrator.
* The orchestrator is responsible for managing everything that happens in Azure, including responding to user requests. Users make requests using the Orchestrator's Web API.
* The Web API can be called by many tools, including the user interface of the Azure portal. When a user makes a request to create a virtual machine, the orchestrator packages everything that's needed. Picks the best server rack and then sends the package and request to the fabric controller. Once the fabric controller has created the virtual machine, the user can connect to it.

## AZURE MARKET PLACE

|  |  |
| --- | --- |
|  | * Azure marketplace offers multiple azure services as a template * For example – if we need a WordPress solution and want to install the solution to one of the VM- we can use the marketplace. * As WordPress is entirely different company which gives a service via Azure marketplace |

# AZURE SUBSCRIPTIONS, MANAGEMENT GROUPS, AND RESOURCES

|  |  |
| --- | --- |
| Screenshot of the hierarchy for objects in Azure. | **RESOURCES**: Resources are instances of services that you create, like virtual machines, storage, or SQL databases.  **RESOURCE GROUPS**: Resources are combined into resource groups, **which act as a logical container** into which Azure resources like web apps, databases, and storage accounts are deployed and managed.  **SUBSCRIPTIONS**: A subscription groups together user accounts and the resources that have been created by those user accounts. For each subscription, there are limits or quotas on the amount of resources that you can create and use. Organizations can use subscriptions to manage costs and the resources that are created by users, teams, or projects.  **MANAGEMENT GROUPS**: These groups help you manage access, policy, and compliance for multiple subscriptions. All subscriptions in a management group automatically inherit the conditions applied to the management group. |

# WORKLOAD

|  |  |
| --- | --- |
|  | * ***Workload is unit functionality which can be an application or service.*** * For example – If we have a web application which we want to host or a database server they are called as “workload”. |

# RESOURCE GROUP

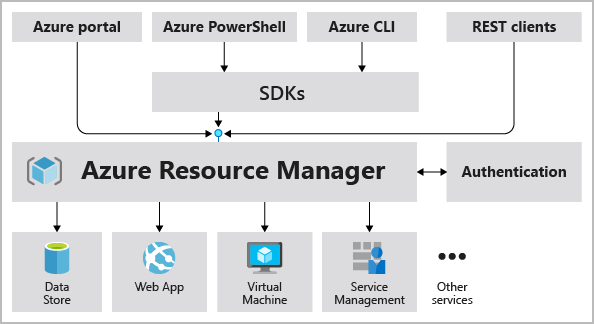
|  |  |
| --- | --- |
| Conceptual image showing a resource group box with a function, VM, database, and app included.  **LIFE CYCLE**   * If you delete a resource group, all resources contained within it are also deleted. * Resource groups make it easy to remove a set of resources all at once.   **AUTHORIZATION**   * Resource groups are also a scope for applying role-based access control (RBAC) permissions. * By applying RBAC permissions to a resource group, we can ease administration and limit access to allow only what's needed. | * Resource groups are a fundamental element of the Azure platform. * **A resource group is a logical container for resources deployed on Azure.** These resources are anything we create in an Azure subscription like VMs, Azure Application Gateway instances, and Azure Cosmos DB instances. * All resources must be in a resource group, and a resource can only be a member of a single resource group. * Many resources can be moved between resource groups with some services having specific limitations or requirements to move. * Resource groups can't be nested. Before any resource can be provisioned, you need a resource group for it to be placed in. |

## CREATING A RESOURCE GROUP

|  |  |
| --- | --- |
| * All the resources are tied to a subscription for billing aspects. * The resource group must associate to a region. * ***A resource cannot be a part of two different resource group*** * ***The resource can only be a part of a one subscription.*** |  |

# AZURE RESOURCE MANAGER

* Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure account. You use management features like access control, locks, and tags to secure and organize your resources after deployment.
* When a user sends a request from any of the Azure tools, APIs, or SDKs, Resource Manager receives the request. It authenticates and authorizes the request. Resource Manager sends the request to the Azure service, which takes the requested action. Because all requests are handled through the same API, you see consistent results and capabilities in all the different tools.



## THE BENEFITS OF USING RESOURCE MANAGER

* Manage your infrastructure through declarative templates rather than scripts. A Resource Manager template is a JSON file that defines what you want to deploy to Azure.
* Deploy, manage, and monitor all the resources for your solution as a group, rather than handling these resources individually.
* Redeploy your solution throughout the development life cycle and have confidence your resources are deployed in a consistent state.
* Define the dependencies between resources so they're deployed in the correct order.
* Apply access control to all services because RBAC is natively integrated into the management platform.
* Apply tags to resources to logically organize all the resources in your subscription.
* Clarify your organization's billing by viewing costs for a group of resources that share the same tag.

# REGION AND ZONES

***A*region*is a geographical area on the planet that contains at least one but potentially multiple datacenters that are nearby and networked together with a low-latency network.*** Azure intelligently assigns and controls the resources within each region to ensure workloads are appropriately balanced.

### WHY REGION IS IMPORTANT

Azure has more global regions than any other cloud provider. These regions give you the flexibility to bring applications closer to your users no matter where they are. Global regions provide better scalability and redundancy. They also preserve data residency for your services.

|  |  |
| --- | --- |
|  | CASE 1: SINGLE DATA CENTER IN A REGION  Imagine that your application is deployed in a data center in London  WHAT WOULD BE THE CHALLENGES?   * Challenge 1: Slow access for users from other parts of the world (high latency) * Challenge 2: What if the data center crashes?   *Your application goes down (low availability)* |
|  | CASE 2: MULTIPLE DATA CENTER IN A REGION  Let's add in one more data center in London  WHAT WOULD BE THE CHALLENGES?   * Challenge 1: Slow access for users from other parts of the world * Challenge 2 (SOLVED) : What if one data center crashes?   *Your application is still available from the other data center*   * Challenge 3: What if entire region of London is unavailable?   *Your application goes down*  16 |
| CASE 3: MULTIPLE REGION – MULTIPLE DATA CENTER    Let's add a new region: Mumbai  WHAT WOULD BE THE CHALLENGES?   * Challenge 1 (PARTLY SOLVED): Slow access for users from other parts of the world   *You can solve this by adding deployments for your applications in other regions*   * Challenge 2 (SOLVED) : What if one data center crashes?   *Your application is still live from the other data centers*   * Challenge 3 (SOLVED) : What if entire region of London is unavailable?   *Your application is served from Mumbai*  ADVANTAGES:   * *High Availability* * *Low Latency* * *Global Footprint* * *Adhere to government regulations* | |

## SELECTING A REGION

* Try to create the VM where majority of the user resides – This will reduce the latency in the response, when user try access the application.
* Another aspect – we need to consider the cost and availability before selecting the region for the service.
* Usually – when we have used across the globe – we make use of Azure CDN service.

## AZURE AVAILABILITY ZONE

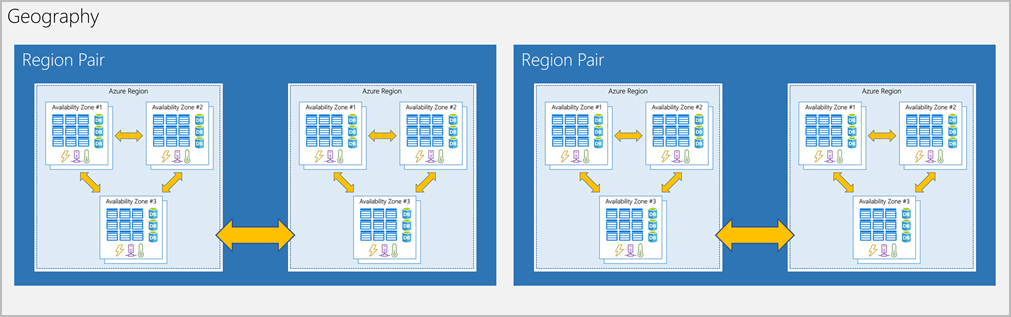
|  |  |
| --- | --- |
| Diagram showing three datacenters connected in a single Azure region representing an availability zone. | * Availability zones are physically separate datacenters within an Azure region. * Each availability zone is made up of one or more datacenters equipped with independent power, cooling, and networking. * An availability zone is set up to be an isolation boundary. If one zone goes down, the other continues working. * Availability zones are connected through high-speed, private fiber-optic networks. |

## AZURE REGION PAIR

Availability zones are created by using one or more datacenters. ***There's a minimum of three zones within a single region***. It's possible that a large disaster could cause an outage big enough to affect even two datacenters. That's why Azure also creates **region pairs**.

Each Azure region is always paired with another region within the same geography (such as US, Europe, or Asia) at least 300 miles away. This approach allows for the replication of resources (such as VM storage) across a geography that helps reduce the likelihood of interruptions because of events such as natural disasters, civil unrest, power outages, or physical network outages that affect both regions at once. If a region in a pair was affected by a natural disaster, for instance, services would automatically failover to the other region in its region pair.

***Examples of region pairs in Azure are West US paired with East US and SouthEast Asia paired with East Asia.***



**ADDITIONAL ADVANTAGES OF REGION PAIRS:**

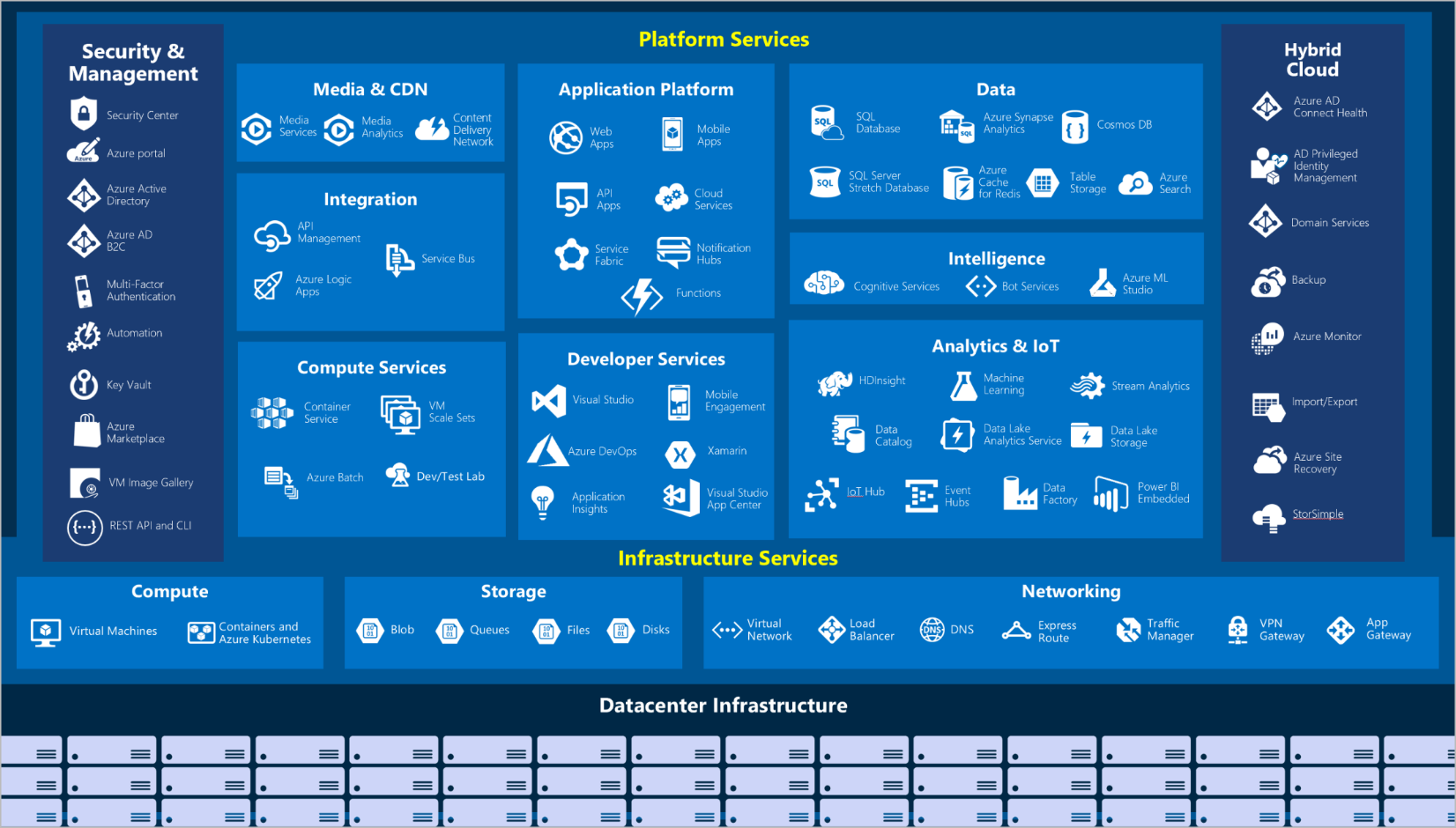
* If an extensive Azure outage occurs, one region out of every pair is prioritized to make sure at least one is restored as quickly as possible for applications hosted in that region pair.
* Planned Azure updates are rolled out to paired regions one region at a time to minimize downtime and risk of application outage.

# AZURE CORE SERVICES

Commomly used categories of Azure Services

1. **COMPUTE**
2. **NETWORKING**
3. **STORAGE**
4. **MOBILE**
5. **DATABASES**
6. **WEB**
7. **INTERNET OF THINGS (IOT)**
8. **BIG DATA**
9. **AI**
10. **DEVOPS**

**Azure Services** : <https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-fundamentals/tour-of-azure-services>



## AZURE COMPUTE SERVICES

* Azure compute is an on-demand computing service for running cloud-based applications. It provides computing resources such as disks, processors, memory, networking, and operating systems. The resources are available on-demand and can typically be made available in minutes or even seconds. You pay only for the resources you use, and only for as long as you're using them.
* Azure supports a wide range of computing solutions for development and testing, running applications, and extending your datacenter. The service supports Linux, Windows Server, SQL Server, Oracle, IBM, and SAP. Azure also has many services that can run virtual machines (VMs). Each service provides different options depending on your requirements. Some of the most prominent services are:

|  |  |
| --- | --- |
| AZURE VIRTUAL MACHINES | With Azure Virtual Machines, you can create and use VMs in the cloud. ***Virtual Machines provides infrastructure as a service (IaaS)*** and can be used in different ways. When you need total control over an operating system and environment, VMs are an ideal choice. Just like a physical computer, you can customize all the software running on the VM. This ability is helpful when you're running custom software or custom hosting configurations. |
| VIRTUAL MACHINE SCALE SET | [Virtual machine scale sets](https://azure.microsoft.com/services/virtual-machine-scale-sets) are an Azure compute resource that you can use to deploy and manage a set of identical VMs. With all VMs configured the same, virtual machine scale sets are designed to support true autoscale. No pre-provisioning of VMs is required. For this reason, it's easier to build large-scale services targeting big compute, big data, and containerized workloads. As demand goes up, more VM instances can be added. As demand goes down, VM instances can be removed. The process can be manual, automated, or a combination of both. |
| AZURE APP SERVICE | With Azure App Service, you can quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. You can meet rigorous performance, scalability, security, and compliance requirements while using a fully managed platform to perform infrastructure maintenance. App Service is a platform as a service (PaaS) offering. |
| AZURE FUNCTIONS (OR SERVERLESS COMPUTING) | Functions are ideal when you're concerned only about the code running your service and not the underlying platform or infrastructure. They're commonly used when you need to perform work in response to an event (often via a REST request), timer, or message from another Azure service, and when that work can be completed quickly, within seconds or less. |
| CONTAINER AND KUBERNETES | [Container Instances](https://azure.microsoft.com/services/container-instances) and [Azure Kubernetes Service](https://azure.microsoft.com/services/kubernetes-service) are Azure compute resources that you can use to deploy and manage containers. Containers are lightweight, virtualized application environments. They're designed to be quickly created, scaled out, and stopped dynamically. You can run multiple instances of a containerized application on a single host machine. |

### VIRTUAL MACHINE

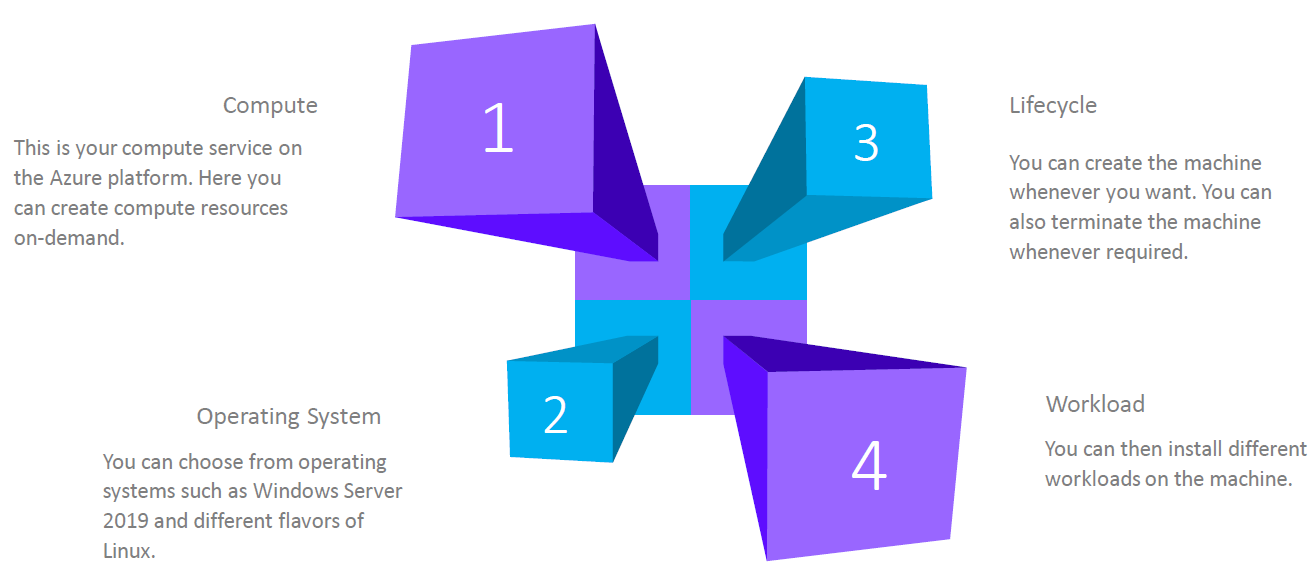
* It is a computer file typically called as an image which behaves like an actual computer.
* This gives you a flexibility that can run multiple machines in a physical computer. Each system can have a different operating system.
* Each of these virtual machines provides its own virtual hardware which includes CPUs, memory, hard drives, network interfaces and other such devices.

#### WHEN TO USE VMs

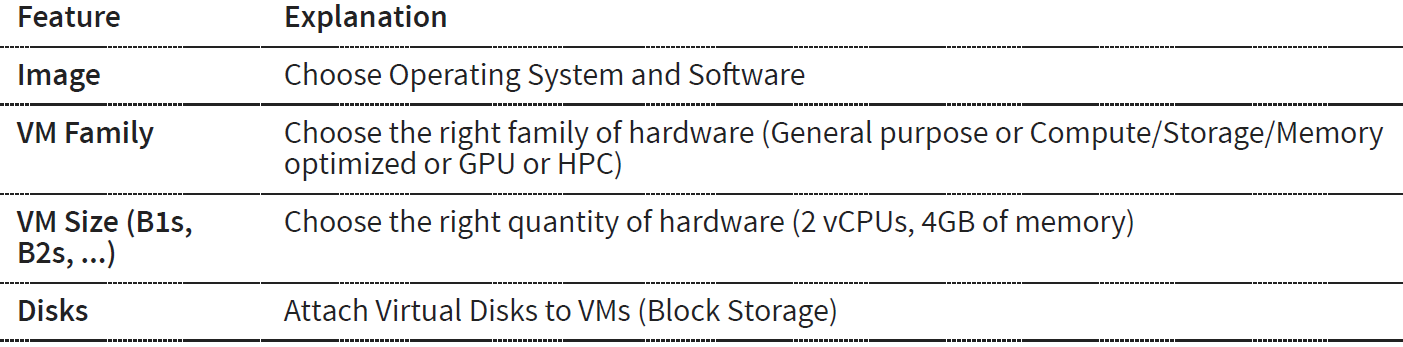
* **During testing and development.** VMs provide a quick and easy way to create different OS and application configurations. Test and development personnel can then easily delete the VMs when they no longer need them.
* **When running applications in the cloud.** The ability to run certain applications in the public cloud as opposed to creating a traditional infrastructure to run them can provide substantial economic benefits. For example, an application might need to handle fluctuations in demand. Shutting down VMs when you don't need them or quickly starting them up to meet a sudden increase in demand means you pay only for the resources you use.
* **When extending your datacenter to the cloud.** An organization can extend the capabilities of its own on-premises network by creating a virtual network in Azure and adding VMs to that virtual network. Applications like SharePoint can then run on an Azure VM instead of running locally. This arrangement makes it easier or less expensive to deploy than in an on-premises environment.
* **During disaster recovery.** As with running certain types of applications in the cloud and extending an on-premises network to the cloud, you can get significant cost savings by using an IaaS-based approach to disaster recovery. If a primary datacenter fails, you can create VMs running on Azure to run your critical applications and then shut them down when the primary datacenter becomes operational again.

#### AZURE’S VIRTUAL MACHINE SERVICE

* In corporate data centers, applications are deployed to physical servers, but we deploy applications in the cloud by renting(provisioning) virtual servers (Virtual Machine)

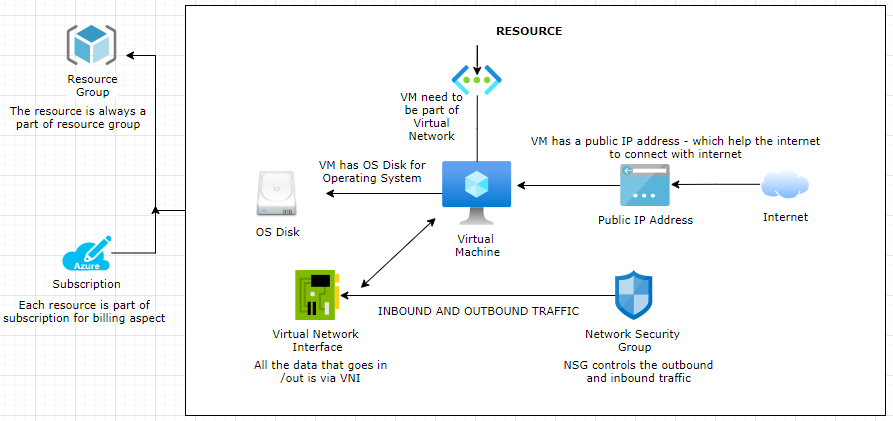


##### AZURE VIRTUAL MACHINE – KEY CONCEPTS

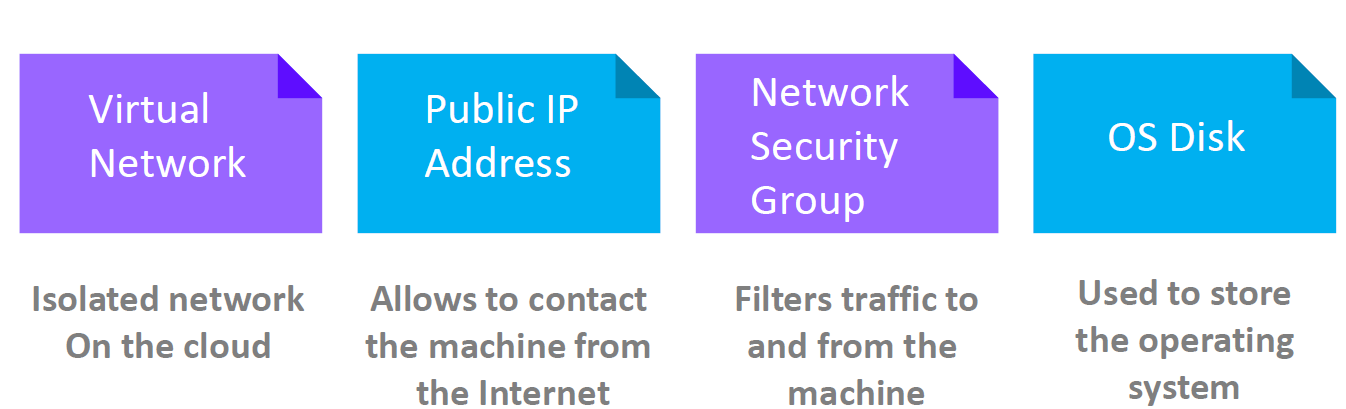


##### DEPLOYING A VIRTUAL MACHINE

* When we deploy a virtual machine – there are other resources also get deployed with it.
* ***VM is a compute resource in Azure Platform. It is an Infrastructure as a service in azure platform***



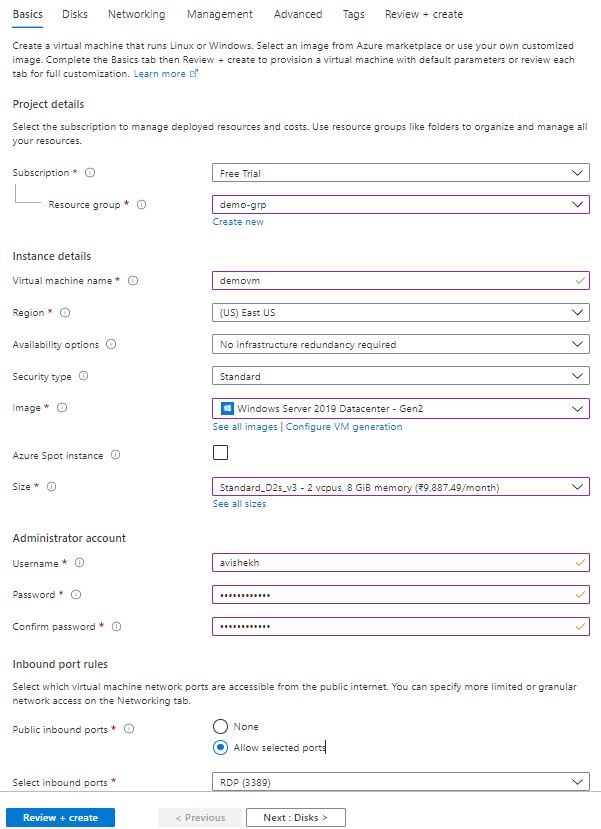
###### RESOURCE DEPLOYED WITH AZURE VM



|  |  |
| --- | --- |
| OS DISK | * The virtual machine has OS Disk where operating system can be installed * Addition disk can be added too |
| VIRTUAL NETWORK INTERFACE | * Virtual Network Interface is like network interface card * All the data that goes in or out go via Virtual Network Interface |
| VIRTUAL NETWORK | * Every VM is part of a Virtual Network. |
| NETWORK SECURITY GROUP | * It controls all the inbound and outbound traffic to and from the VM |
| PUBLIC IP | * The VM are always associated to a public ip address –through which the internet can connect with the VM |

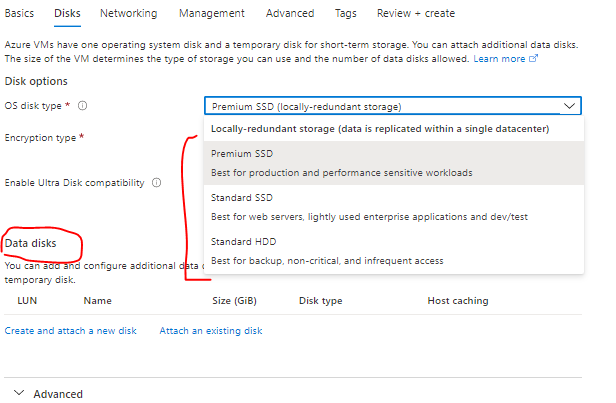
###### DEPLOYING WINDOWS VM

BASIC



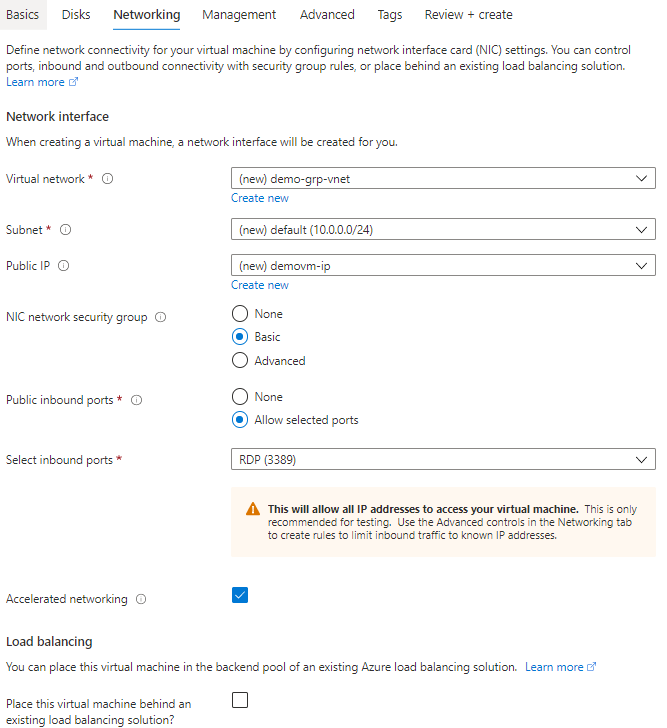
|  |  |
| --- | --- |
| RESOURCE GROUP | The resource group of the VM |
| VM NAME | Name of the Virtual machine |
| REGION | Region in which this VM will be created |
| IMAGE | This is the name of the image – which will be used to set-up OS in the VM. |
| SIZE | This defines the size of the resources in the VM like CPUs and RAM |
| SELECT INBOUND PORTS | This defines on which port the inbound traffic to VM is allowed. For window we user RDP(Remote Desktop) at port 3389 |

DISK



|  |  |
| --- | --- |
| OS DISKTYPE | * This is the disk which will be used for the storage of data |
| DATA DISK | * When we configure the VM – by default we get the OS disk . For additional disk we can add as a Data disk to the VM |

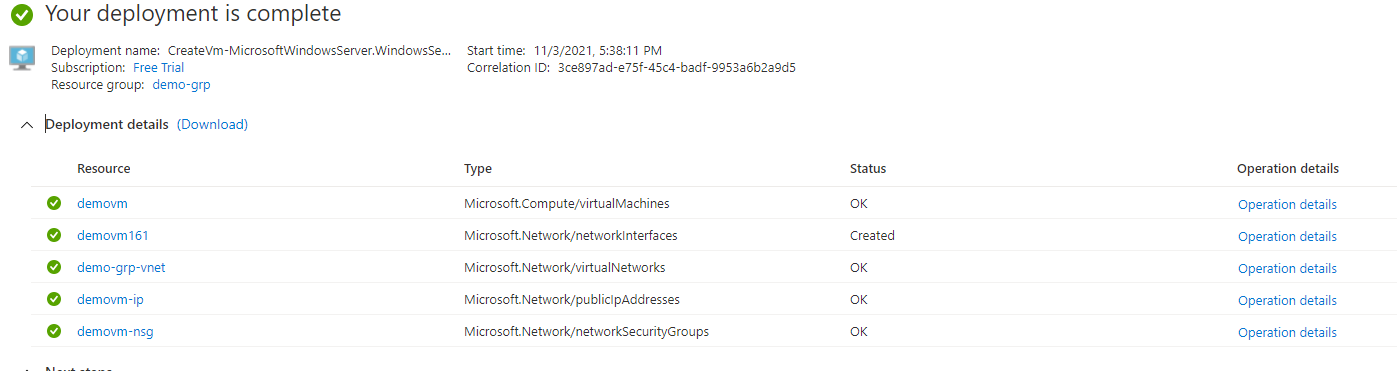
NETWORKING



|  |  |
| --- | --- |
| VIRTUAL NETWORK |  |
| SUBNET |  |
| PUBLIC IP ADDRESS | The VM can be accessed on Internet using its public IP address |

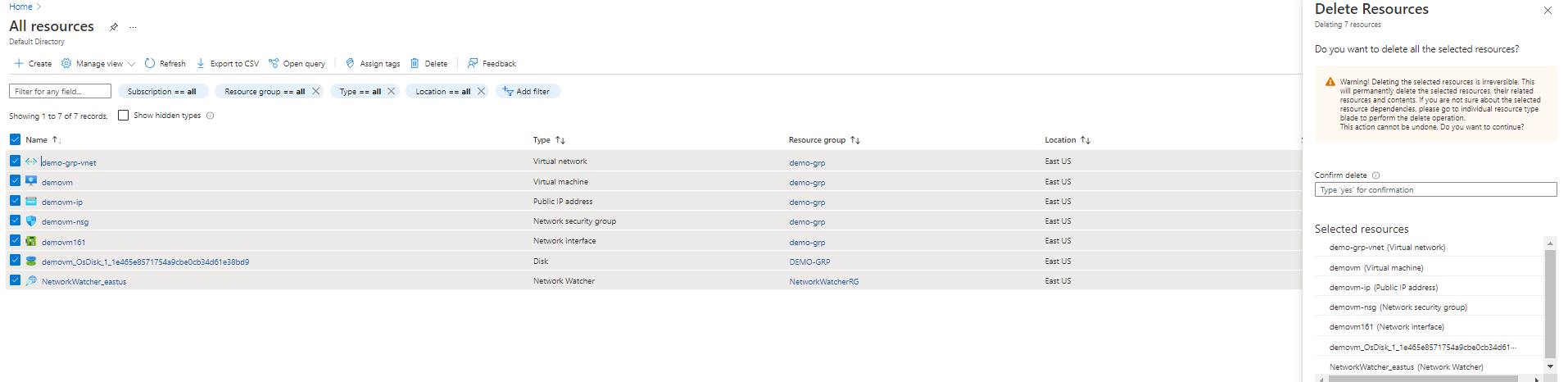
VM CREATED

* The below diagram shows the created VM and other related resources which are created with the VM like NIC, VM , NSG and public IP address.



DELETING A VM

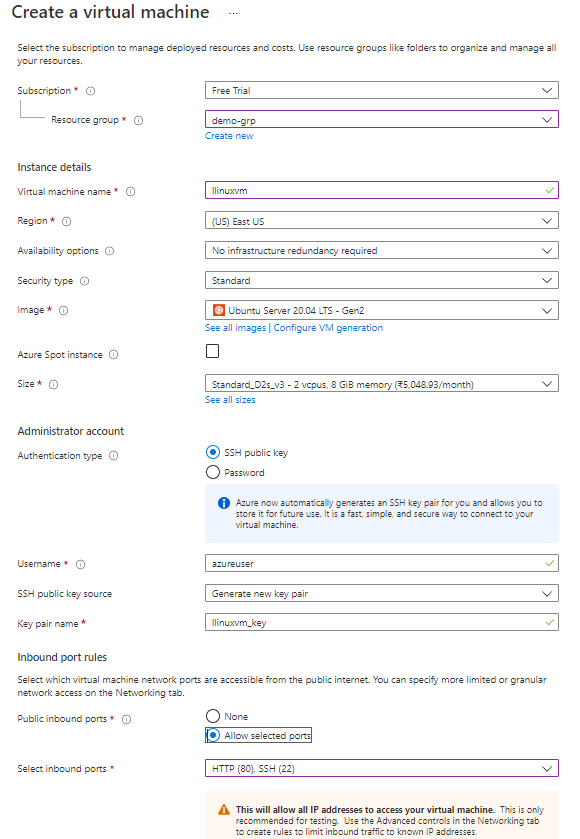
* From the *menu 🡪 All Resources*. Select all the resources which are related to the VM 🡪 Type yes and Delete



###### CONNECTING TO VM (WINDOWS) USING RDP

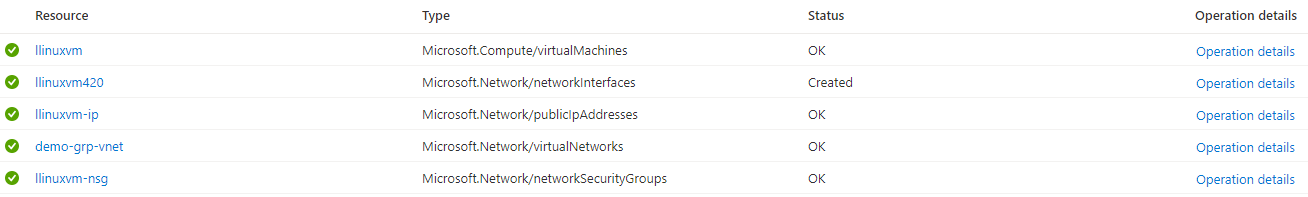
|  |  |
| --- | --- |
|  | * To connect with Windows based VM we use RDP * SSH is used to connect to Linux based VM |
|  | * To connect to the windows VM- download the RDP file * Open the downloaded RDP and Enter the username/password (used while creating the VM) |

DEPLOYING LINUX VM



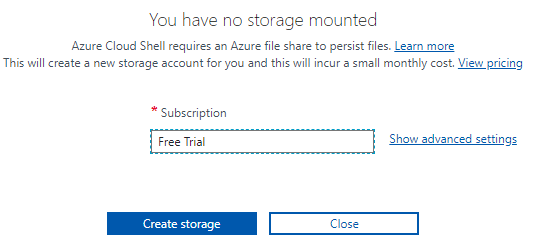
|  |  |
| --- | --- |
|  | * This will show a pop- up to download the SSH key * Download the SSH key – This key will be used to login to VM using SSH. * Note to connect with Linux VM we use SSH. |

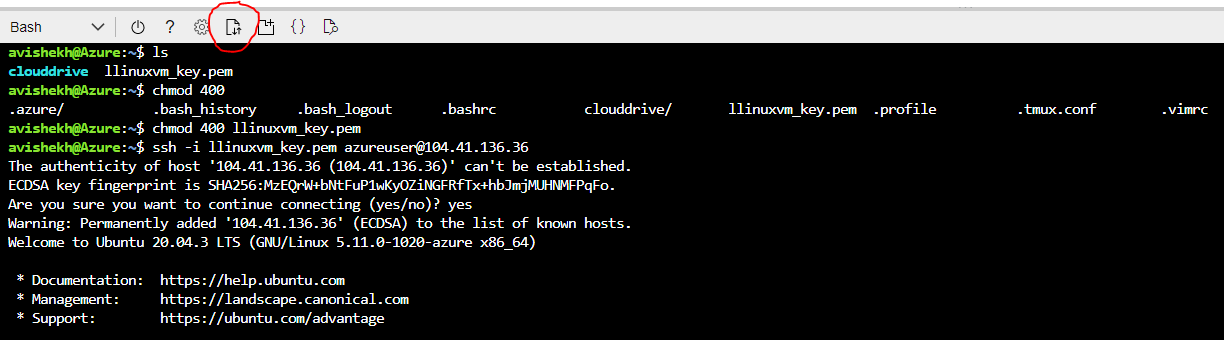
**RESOURCES CREATED WITH VM**



###### CONNECTING TO VM (SSH)

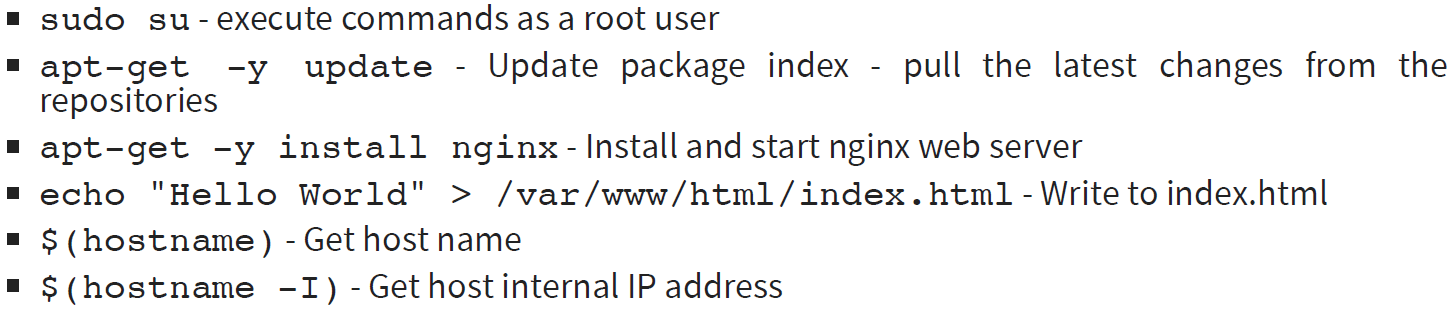
* Open the Cloud Shell and choose Bash
* Select the subscription 🡪 Create Storage



* Upload the SSH key using upload option 
* Navigate to VM 🡪 Connect 🡪SSH . Follow the steps to SSH the VM

###### INSTALLING SOFTWARES IN VM

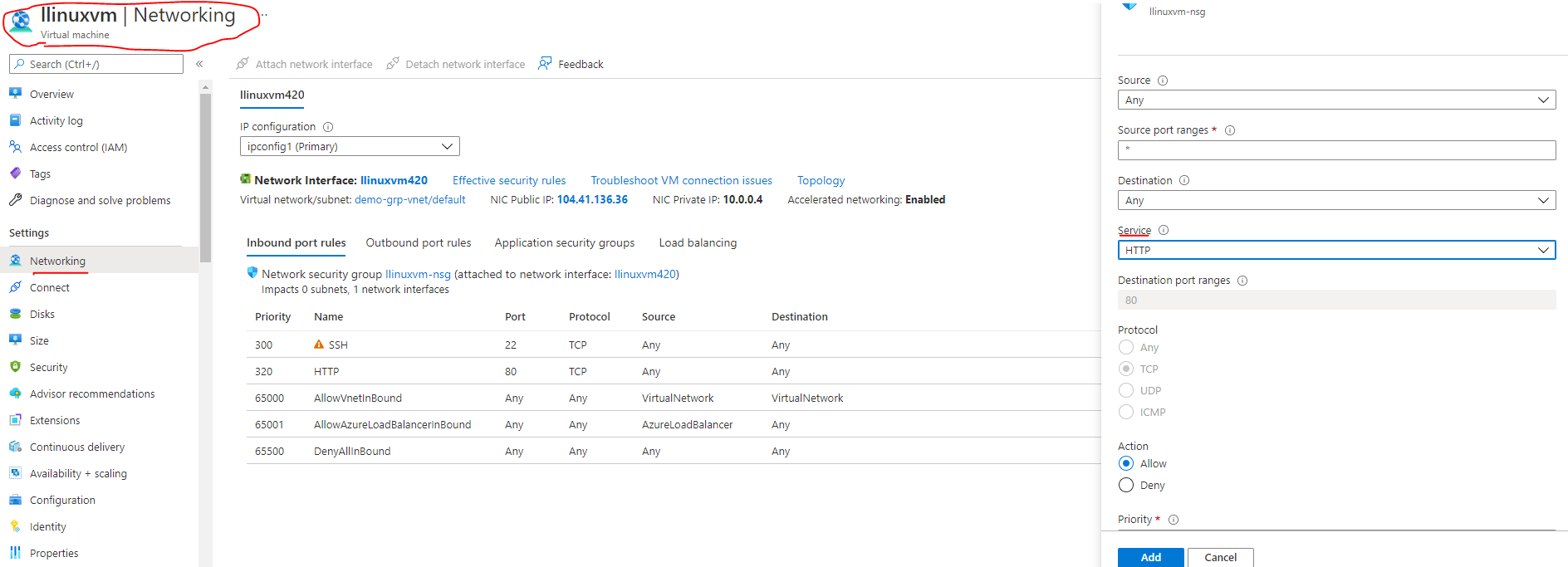
INSTALLING NGINX (HTTP SERVER)



* The nginx webserver can be accessed using the public IP address of the VM(<http://104.41.136.36/> )

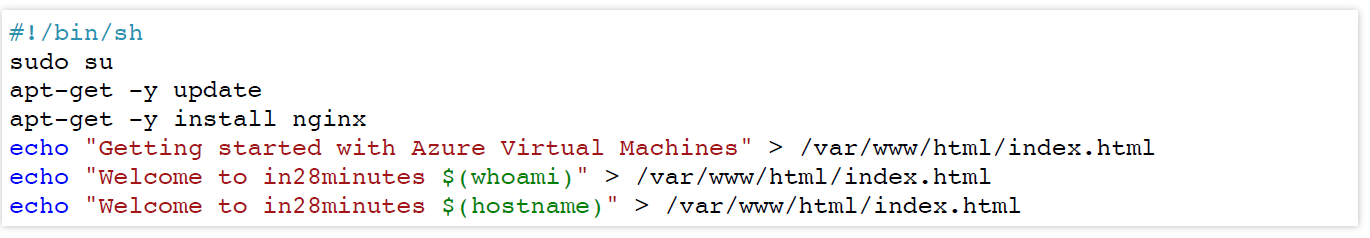
ADDING INBOUND PORT RULE

* We can add the inbound port rule to accept the incoming request. For example – we can enable the HTTP service at port 80 by adding a new inbound port rule.
* The same can be done while creating the VM (***Basic 🡪 Select inbound port***)



USING CLOUD INIT

* In the above steps – We first created the VM and the installed the nginx server. We did all by doing the “***ssh***” and running command from cloud shell.
* Just In case if we want to run a specific set of command after the VM start up – we can write the series of command in “Advanced Tab” 🡪 Cloud init as a bash script. As show below.



QUOTA OF VM

* We do have a limit on number of VM we can create the number of VM we can create. The limitation is always tied to the type of subscription we have.
* Navigate to Dashboard 🡪 Subscription 🡪 Open the subscription 🡪 Usage and Quota



* Note – when we delete the resource group – this will delete all the resources in that resource group.

#### SCALE VMS IN AZURE

For high availability, scalability, and redundancy of VMs, azure has several features that can meet them. These features include:

* + VIRTUAL MACHINE SCALE SETS
  + AZURE BATCH

##### WHAT IS AZURE BATCH?

Azure Batch enables large-scale parallel and high-performance computing (HPC) batch jobs with the ability to scale to tens, hundreds, or thousands of VMs. When you're ready to run a job, Batch does the following:

1. Starts a pool of compute VMs for you.
2. Installs applications and staging data.
3. Runs jobs with as many tasks as you have.
4. Identifies failures.
5. Requeues work.
6. Scales down the pool as work completes.

There might be situations in which you need raw computing power or supercomputer-level compute power. Azure provides these capabilities.

##### VIRTUAL MACHINE SCALE SETS

* Virtual machine scale set simplify the creation and management of multiple VMs (group of VMs)
* VM scale set also allow as to add a load balancer
* Supports manual and auto scaling
* Distribute VM instances across multiple Availability Zones
* Supports 1000 VM instances in a single scale set.

###### EXAMPLE

* Let's say we have hosted an application, or you have some sort of workload that needs to be hosted on a virtual machine. If that the application is putting high load on the virtual machine- We can solve this issue one alternative can be – To change the size of the virtual CPU. but there is a limit on how much we can scale the VM. And sometimes your application might not behave properly just by increasing the number of virtual CPUs or the memory.
* Other alternative can be - To host the application on multiple VMs and then we increase or scale the number of machines, when multiple hit the application. Here the request will be distributed across the virtual machines.
* ***Virtual machine skill set so this service can be used to scale the number of machines. This is different from***

***distributing the traffic across the machines. That is done by another service that is known as the load balancer.***

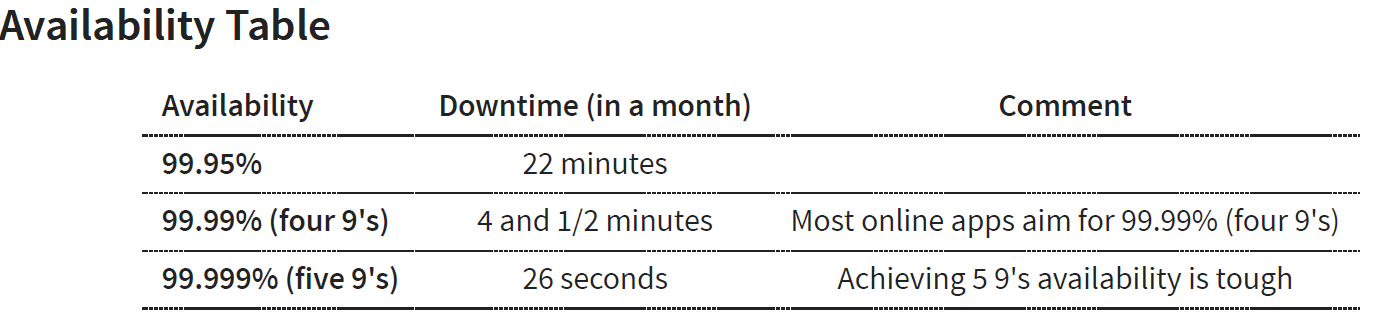
* Hence using VM scale set service we can provision based on demand identical virtual machines.
* So initially, we can have one machine that has been created by the virtual machine scale set. Then we can define rules. For example – it can say - scale out if the CPU percentage goes beyond 70%.
* So If it goes beyond 70%, then add one more machine and now we'll have two machines as part of our infrastructure.
* **Note: We must ensure that our application is installed on both machines. This is not going to be done automatically by the scale set service.**

###### SCALLING IN VM SCALE SET

1. At peak demand /load. We can have three machines that have been spun up. If we have three machines, we are billed for all three machines
2. But now the demand of the application has become less. According to the scale set rule(scaling policy) on scale set it will be scaled down, let’s say 1 VM – Then we will be charged for just for 1 VM.

###### CREATING A VM SCALE SET

#### AVAILABILITY



##### INCREASING THE AVAILABILITY

* **SINGLE INSTANCE VM:** If we are using single instance VM we can use a specific disk type . Below is the availbility – when we select a spefic disk type(from Disk Tab)

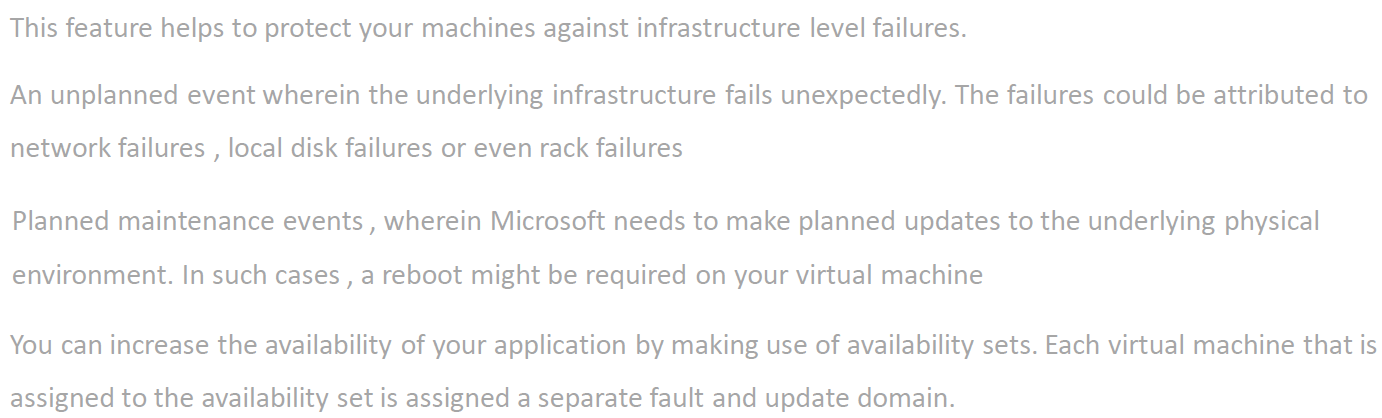
|  |  |
| --- | --- |
| **DISK TYPE** | **AVAILABILITY** |
| Premium SSD or Ultra Disk | 99.9% |
| Standard SSD Managed Disks | 99.5% |
| Standard HDD Managed Disks | *95%* |

Two or more instances in same Availability Set: 99.95%

* Availability set is a logical grouping of VMs
  + **FAULT DOMAINS**: Group of VMs sharing a common power source and network switch. We can create upto 3 fault domains
  + **UPDATE DOMAINS**: Group of VMs that are rebooted (updated) at the same time.We can create up to 20 update domains
* Two or more instances in two or more Availability Zones in the same Azure region: 99.99%
* ***Summary: Create multiple instances in multiple AZs if you want high availability***

#### AVAILABILITY SETS

AVAILIBILITY SET AND ZONES ARE CREATED TO INCREASE THE AVALIBILITY OF THE APPLICATION DEPLOYED TO VMS

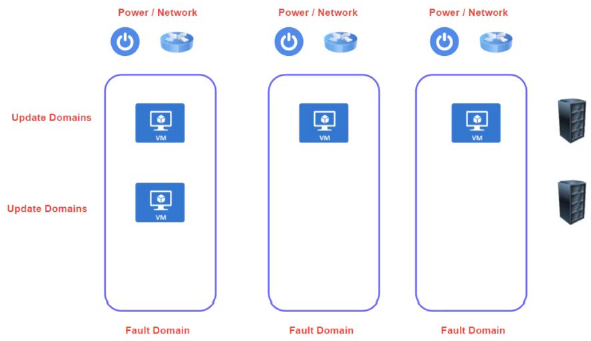


|  |  |
| --- | --- |
|  | * When we deploy the application is multiple VMs – these VM are actually created in phycial server in Azure Data Center. * The phycial server has its dedicated power source and networking. * When we spin-up a VM – we cannot control in which phycial server the VMs are created. * Just in case the physical server goes down (may be due the power source) , then both VMs will go down and hence the application. * To solve this option Azure platform has offers “Availability Set” |

* When a VM is created it is configured to be part of Fault Domain and Update Domain.
* This feature helps to protect your machines against infrastructure level failures.
* An unplanned event wherein the underlying infrastructure fails unexpectedly. The failures could be attributed to network failures , local disk failures or even rack failures
* Planned maintenance events , wherein Microsoft needs to make planned updates to the underlying physical environment. In such cases , a reboot might be required on your virtual machine
* You can increase the availability of your application by making use of availability sets. Each virtual machine that is assigned to the availability set is assigned a separate fault and update domain.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **UD1** | **VM1** | **VM3** | | **UD2** | **VM2** | **VM4** | |  | **FD1** | **FD2** | | * In the following matrix diagram – Virtual Machine VM1 & VM2 belong to a Fault Domain (FD1)and VM3 and VM4 belong to fault domain FD2 * VM1 and VM3 belong to update domain UD1 and VM2 an VM4 belong to update domain UD2 |

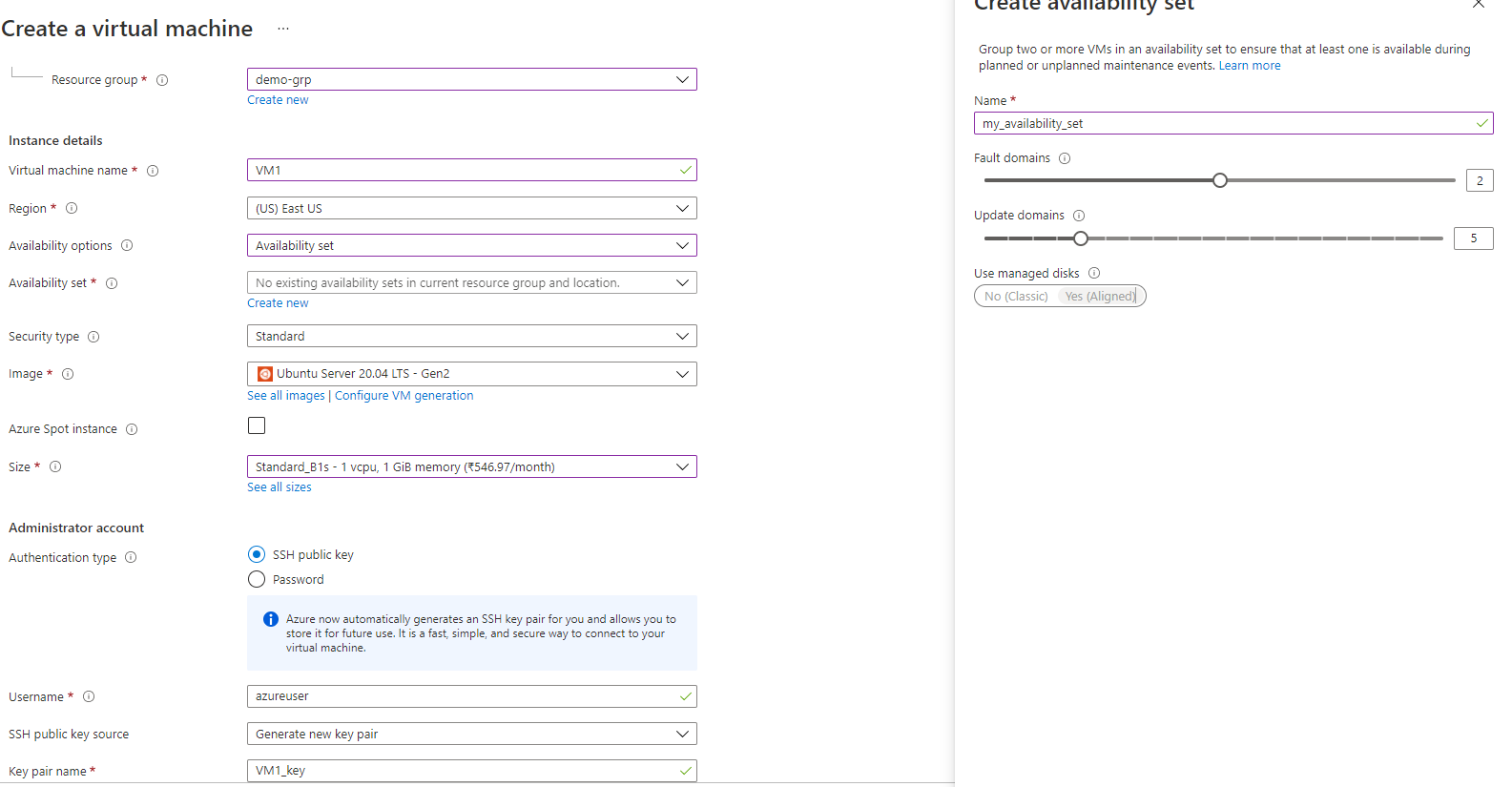
* As the fault domain shares the common power source and network switch – Just in case some goes wrong with this fault domain – then the application will be still available via VM3 and VM4 and vice versa.
* When the physical server needs an update – It will be updated based on update domain. Hence – if update domain UD1 is getting updated then applicatiomn will be available via VM2 and VM4.



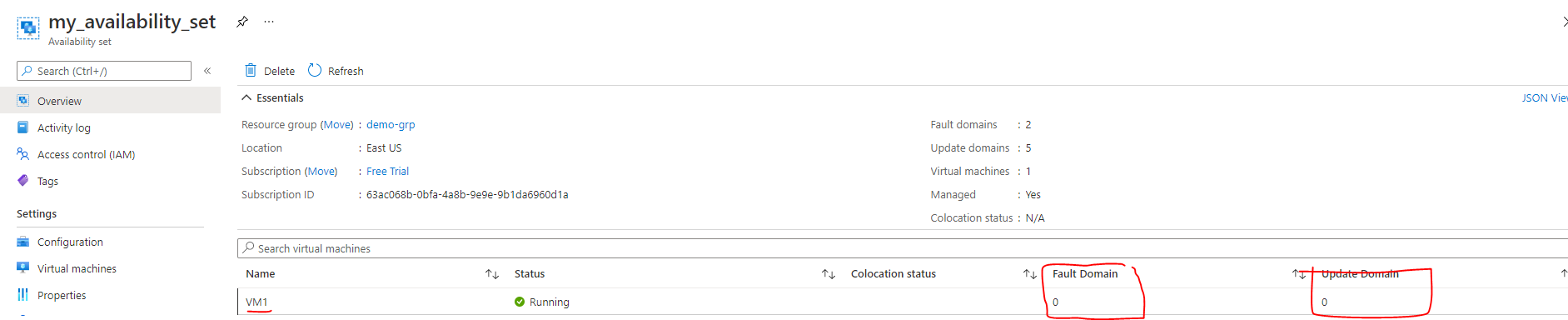
##### EXAMPLE

* Let’s create 2 VMs and make them a part of a availability set. Here Azure will make sure that VMs are properly distributed among fault and update domain.

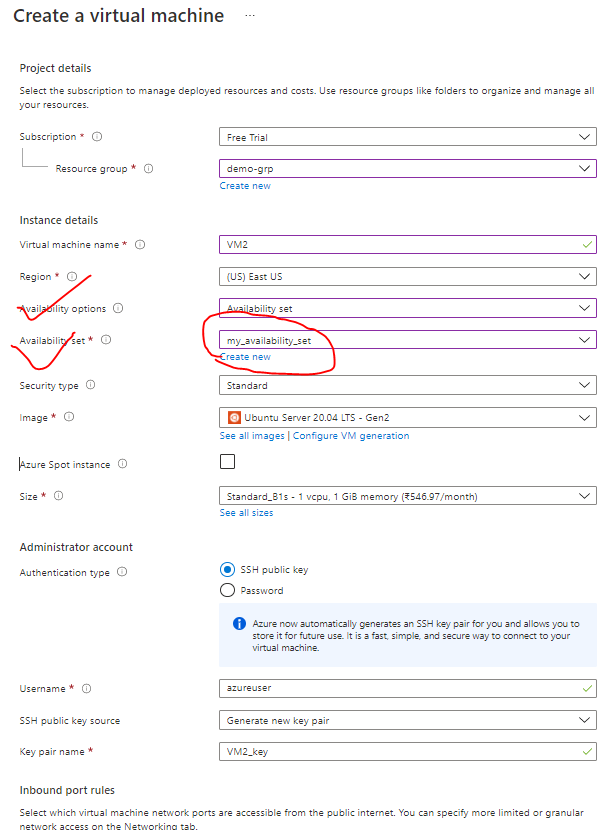
*CREATING THE FIRST VM*



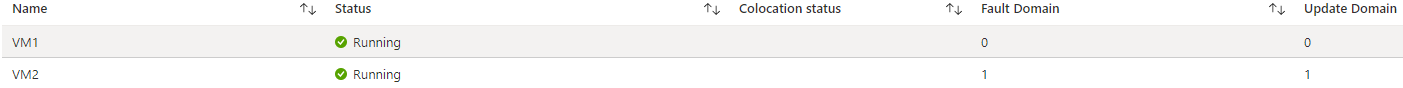
*AVAILABILITY SET STATUS:* The first VM is created in Fault Domain = 0 and update Domain= 0



*CREATING THE SECOND VM IN THE SAME AVAILABILITY SET*



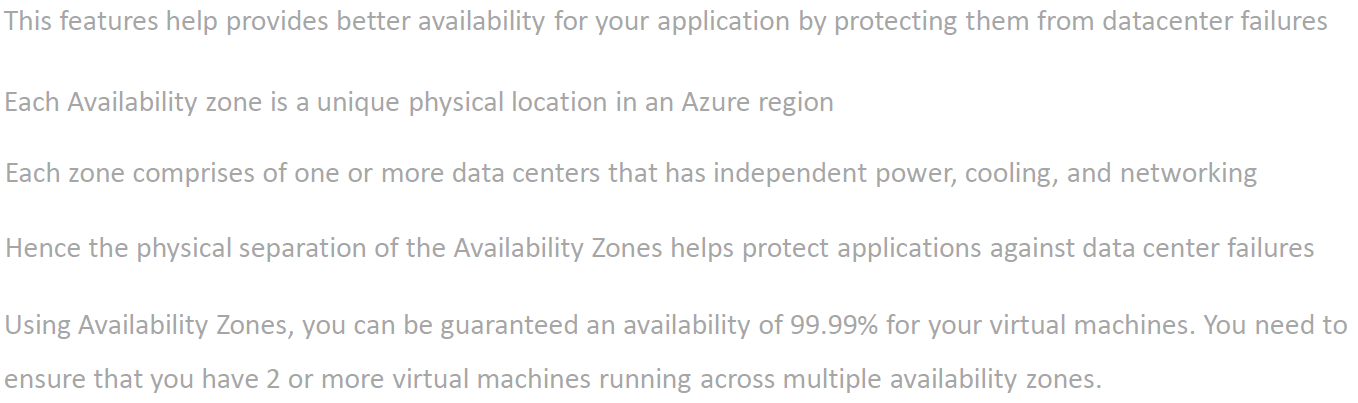
*AVAILABILITY SET STATUS:* The second VM is created in Fault Domain = 1 and update Domain= 1



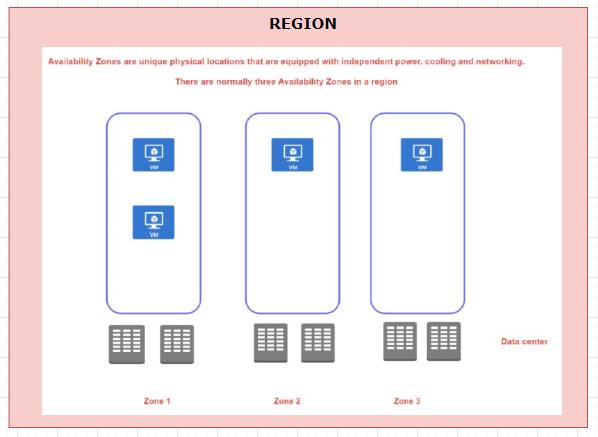
* Hence Azure makes sure that VMs are evenly distributed between the Fault and update domain in a availability zone.
* The configuration of availability set for VM can only be done while creating the VM. VM belong to another availability set cannot eb migrated to. another availability set

**Note: The application deployed to a VM needs to be synced to another VM manually even it belong to same availability set.**

#### AVAILABILITY ZONES



* In a specific region – we have multiple zones and each zone is a collection of data centers.
* Availability zones are unique physical location that are equiped with independent power , colling and networking.
* This features help provides better availability for your application by protecting them from datacenterfailures
* Each Availability zone is a unique physical location in an Azure region
* Each zone comprises of one or more data centers that has independent power, cooling, and networking
* Using Availability Zones, you can be guaranteed an availability of 99.99% for your virtual machines. You need to ensure that you have 2 or more virtual machines running across multiple availability zones.



* When we create a VM in an availability zone – It is in turn mapped to the data center in that availability zone.
* The advantage we get with availability zone is that – if a data centers in a zone goes down – the application will be up and running from other zones in a specific region.

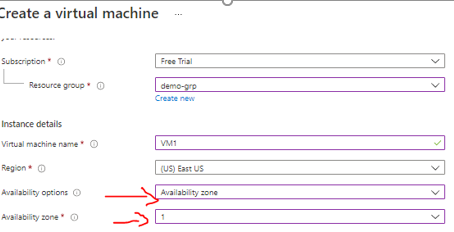
##### NOTES ON AVAILABILITY ZONES AND SET

* There is no extra cost involved in creating availability zone or sets. But the is a costing aspect – when it comes to VM communication between the Zones. But this cost is not applicable when it comes to availability set as those VM are part of same physical data center.
* When we create a VM is availability zone or set – it our responsibility to sync the application in all the newly created VMs- To achieve the sync we can make use of extension of custom script called “Cloud Init Script”.

##### EXAMPLE

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | * For a region we have availability zone. * Lets create 2 VMs in two different availabilty zones in a same region.  |  |  |  | | --- | --- | --- | | VM | REGION | AVAILABILITY ZONE | | VM1 | EAST US | 1 | | VM2 | EAST US | 2 | |

CREATING THE VM IN AVAILIBILITY (ZONE -0 ; REGION – EAST US)



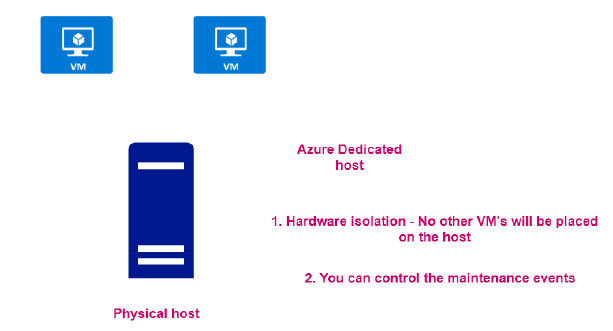
**VMS IN DIFFERENT AVAILIBILITY ZONE**

|  |  |
| --- | --- |
| VM1- OVERVIEW | VM2- OVERVIEW |
|  |  |

**WHY SHOULD I CREATE AVAILABILITY SETS IF WE HAVE AVAILABILITY ZONE?**

* When we create VM in different availability zone, there is an extra cost involved – which is called bandwidth pricing (<https://azure.microsoft.com/en-us/pricing/details/bandwidth/> ).
* Hence, we need to consider the bandwidth pricing while creating the VM is different availability zones.
* Bandwidth Price involve the charges on the data transfer between VM on different zones, especially when it comes to data used while syncing the application between VMs

#### AZURE DEDICATED HOST



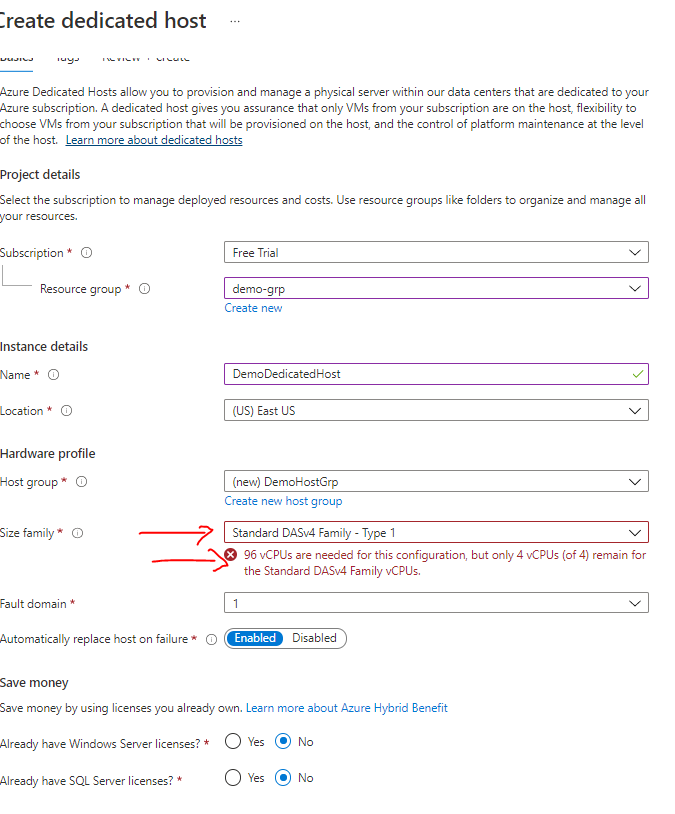
* When we create VMs, they are get created in physical server/ host in the azure data center
* In Azure – it come with a capability to assign the entire physical host as a resource. Note this capability are usually leveraged by large scale organization.

##### ADVANTANGE OF DEDICATED HOST

* As it is a dedicated host – no other VM can be placed in the host.
* We can be able to control the maintenance events of the VMs

##### CREATING DEDICATED HOST

* Navigate to Market Place 🡪 Search “Dedicated Host”.
* The creation of Dedicated host failed in below diagram as it need more resources (i.e 96 CPU) – which are not applicable for free subscription.



## AZURE NETWORK SERVICES

### AZURE VIRTUAL NETWORK

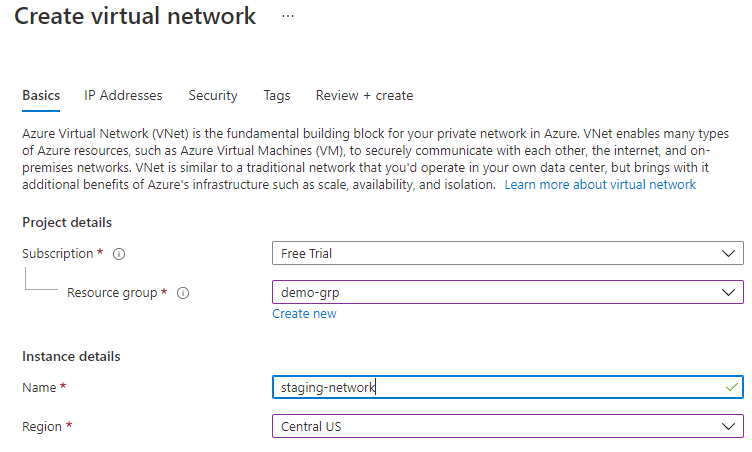
* The Virtual network is a is always a range of IP address.
* When we create a VM it is always get a private IP address. If the private IP of the VM fall in the range of virtual machine, then the machine will be part that virtual network. Hence, The private IP helps in locating the VM in the vitual network.
* The Azure Virtual Network service is used to define an isolated network in Azure. The virtual network can then be used to host the resources such as Azure virtual machines.
* The Azure virtual network gets assigned an address space (IP address range) which we can specify when we create an Azure virtual network.

#### CREATING A VIRTUAL NETWORK

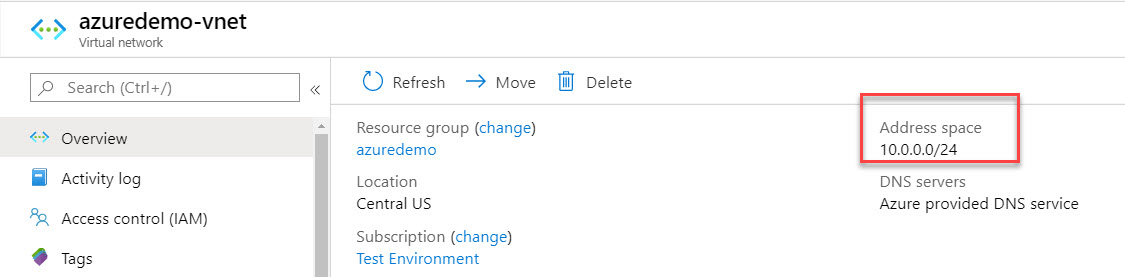
* Usually the Virtual network is created aling with the VM. But we can also able to create Virtual machine upfront and the tie the VM into it.



* ***Step 1***: Lets create a Virtual al network. This wizard will create subnet as well in the SUBNET as well network.



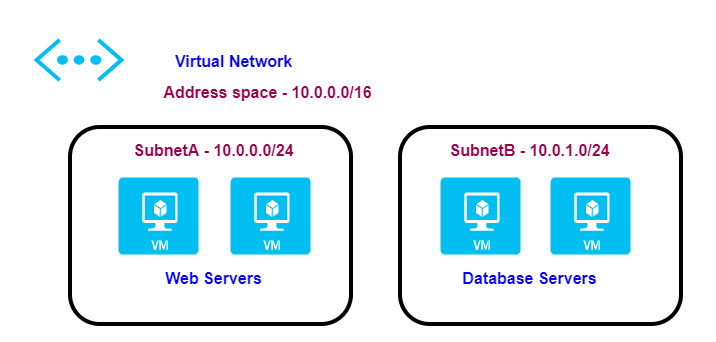
* *Note: When we create a VM needs to be part of virtual network*
* Virtual network has something called IP address range / address space as shown below

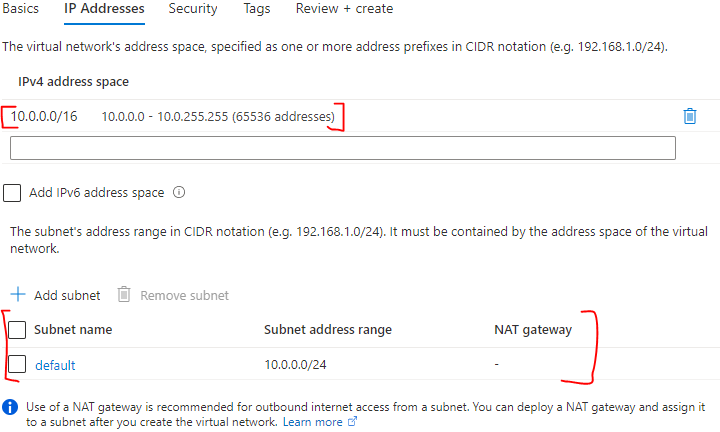


#### CREATING A SUBNET IN VIRTUAL NETWORK

After creating a Virtual Network – We can then add subnets to the Azure virtual network. This helps divide the network into more logical segments.

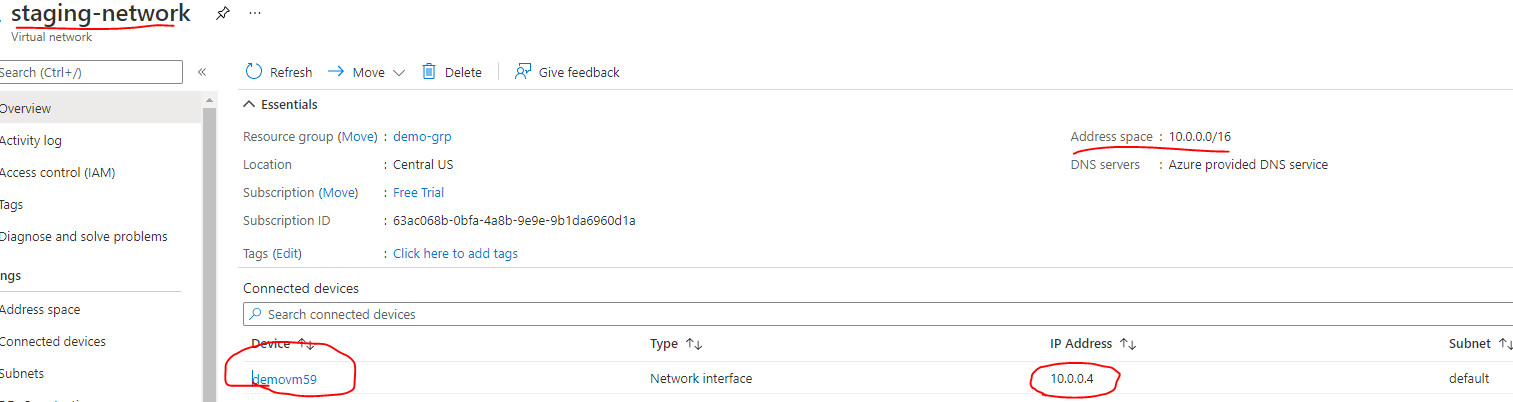
An example is shown below of having multiple subnets. You could have one subnet named SubnetA in the virtual network to host your Web servers and another subnet to host the Database servers.





#### CREATING A VIRTUAL MACHINE IN VIRTUAL NETWORK

* While creating a VM we need in a virtual network – we need to select the same region as of the Virtual Network.
* If the correct region is selected in the “Basic” tab – then only the Virtual network will be visible in “Networking” tab.



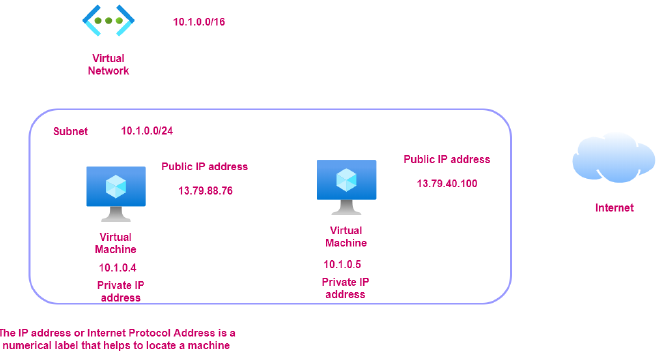
* The above diagram shows the VM created in the virtual network.
* The IP of the VM lies within the range of Virtual Network

*Questions*

1. *If I have pre-created VM (which belong to some another virtual network) can I change the Virtual network of the VM ? - No*
2. *Can a VM can be a part of two different Virtual Network? – No*

#### COMMUNICATION ACROSS VIRTUAL MACHINES IN A VIRTUAL NETWORK

* The virtual machines within the Virtual network can communicate using their private IP address.

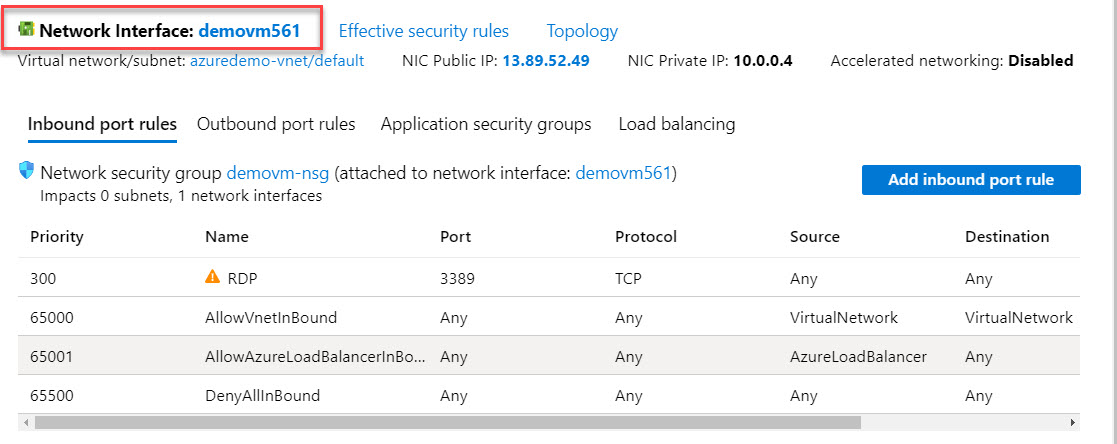


### NETWORK SECURITY GROUP

* Network security group is used to filter the in-bound and outbound traffic which is flowing to the VM.
* All the data flows into VM go through the Virtual Network Interface. Hence – when we access the VM using the public IP address. The traffic flow via Virtual Network interface.
* The Network security group (which is attached to the Virtual Network Interface) - has set of rules which controls / filters the inbound and outbound traffic. It’s like a basic firewall.
* The NSG rules can also be applied on subnet layer as well.
* All the inbound and outbound rules can be set from “Networking” in VM dashboard.



* By default, all traffic into a virtual machine is DENIED. We must explicitly add rules to allow traffic into a virtual machine
* There are also outbound rules to control the traffic flowing out of the virtual machine. By default, all traffic outbound onto the Internet is *allowed*.
* The inbound / outbound rules include the rules around protocol, port number, Source and Destination and Priority.
* The Network security rule are evaluated from top to bottom. If the rule matches in rule, rest of the rules will be ignored



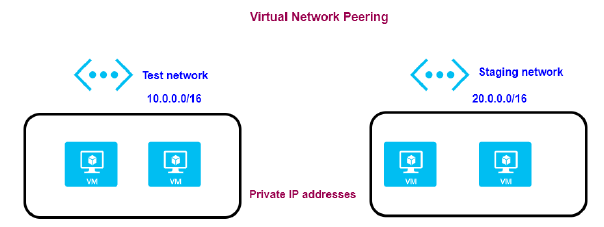
#### APPLICATION SECURITY GROUP

|  |  |
| --- | --- |
|  | * Let’s say – we have VM which has webserver which is in turn communicating with a VM having DB server. * Application Security Group is one of the ways of filtering traffic using IP address. * In the above example if we have multiple VM(webserver) communicating with the DB server. We create an Application Security Group – which has list of all such VMs. * At the DB VM side – we have to make sure that its NSG should accept the incoming request from the Application Security Group- which in turn means DB VM is allowing the incoming request from webservers |

### NETWORK CONNECTIVITY OPTIONS

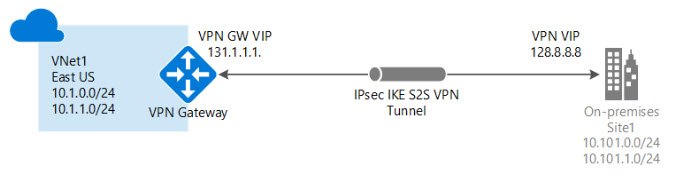
#### **VIRTUAL NETWORK PEERING**

* Virtual Network Peering is used to connect two Azure virtual networks together via the backbone network.
* Azure supports connecting two virtual networks located in the same region or networks located across regions.
* Once we enable virtual network peering between two virtual networks, the virtual machines can then communicate via their private IP addresses across the peering connection.
* We can also peer virtual networks that are located across different subscriptions.
* The virtual networks can't have overlapping CIDR blocks.



#### **SITE-TO-SITE VPN CONNECTION**

* A Site-to-Site VPN connection is used to establish a secure connection between an on-premise network and an Azure network via the Internet.



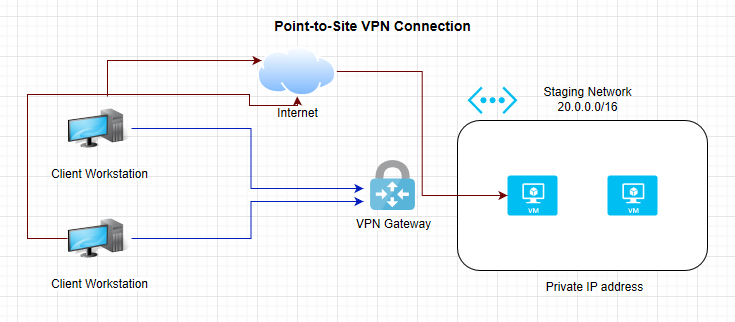
* On the on-premise side, you need to have a VPN device that can route traffic via the Internet onto the VPN gateway in Azure. The VPN device can be a hardware device like a Cisco router or a software device ( e.g Windows Server 2016 running Routing and Remote services). The VPN device needs to have a publically routable IP address.
* The subnets in your on-premise network must not overlap with the subnets in your Azure virtual network
* The Site-to-Site VPN connection uses an IPSec tunnel to encrypt the traffic.
* The VPN gateway resource you create in Azure is used to route encrypted traffic between your on-premise data center and your Azure virtual network.

#### **POINT-TO-SITE VPN CONNECTION**

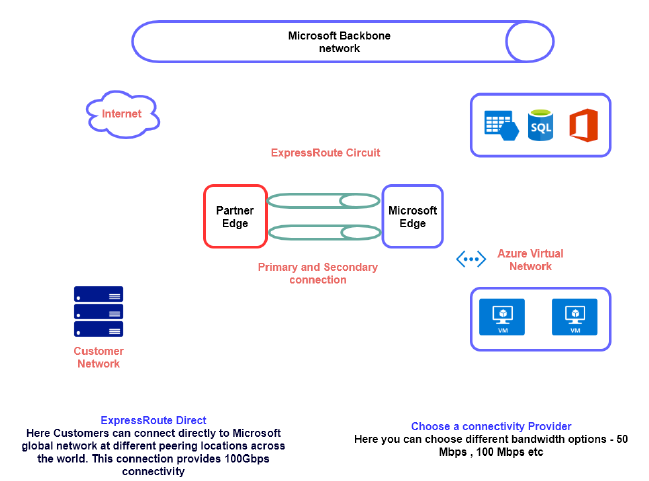
* A Point-to-Site VPN connection is used to establish a secure connection between multiple client machines and an Azure virtual network via the Internet.

##### USE CASE

* If we have multiple Client workstation which needs a connection to the Virtual Network.
* Usually, the VM can be accessed using the public IP address.
* Since the connections is via internet hence for security standpoint – the client workstations should talk to VM via private IP address. For the communication via Private IP address with the VM , the client workstation has to connect via VPN.
* For enable this set-up we need to deploy a resource called VPN gateway resource and configure Point to site VPN connection.
* In addition to that the set up make use of certificates for authentication to make the connection more secure



#### AZURE EXPRESS ROUTE



## AZURE STORAGE SERVICE

* A storage account provides a unique namespace for your Azure Storage data, that's accessible from anywhere in the world over HTTP or HTTPS. Data in this account is secure, highly available, durable, and massively scalable.
* Azure storage account provides the storage on cloud and we use a Azure storage service account for the same.
* Azure Storage account is a service – but based on type of storage account we can make use of different storage/services
* Azure Storage is also used by infrastructure as a service virtual machine, and platform as a service cloud services.

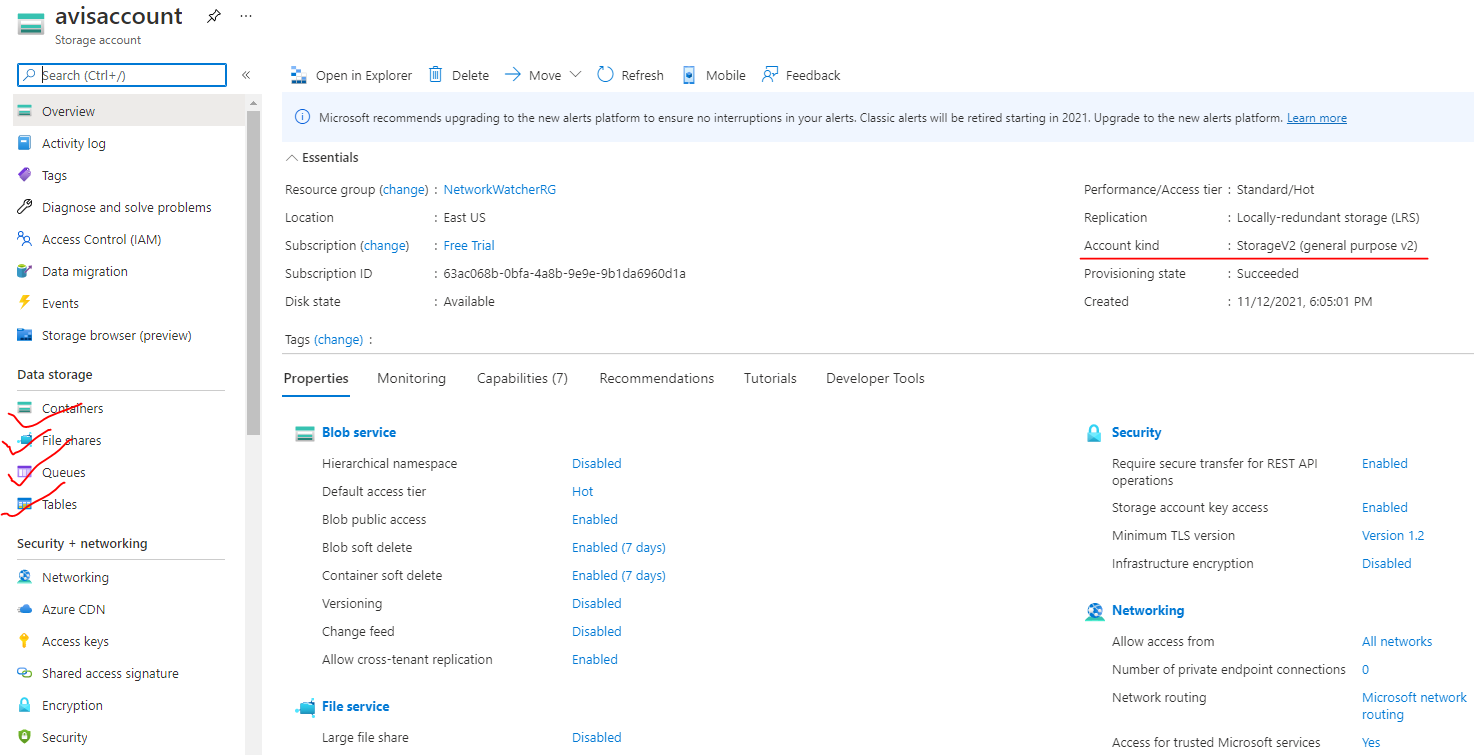
### CREATING A STORAGE ACCOUNT (GENERAL PURPOSE V2 STORAGE ACCOUNT)

* Create Resource 🡪 Storage Account
* Give the storage Account name

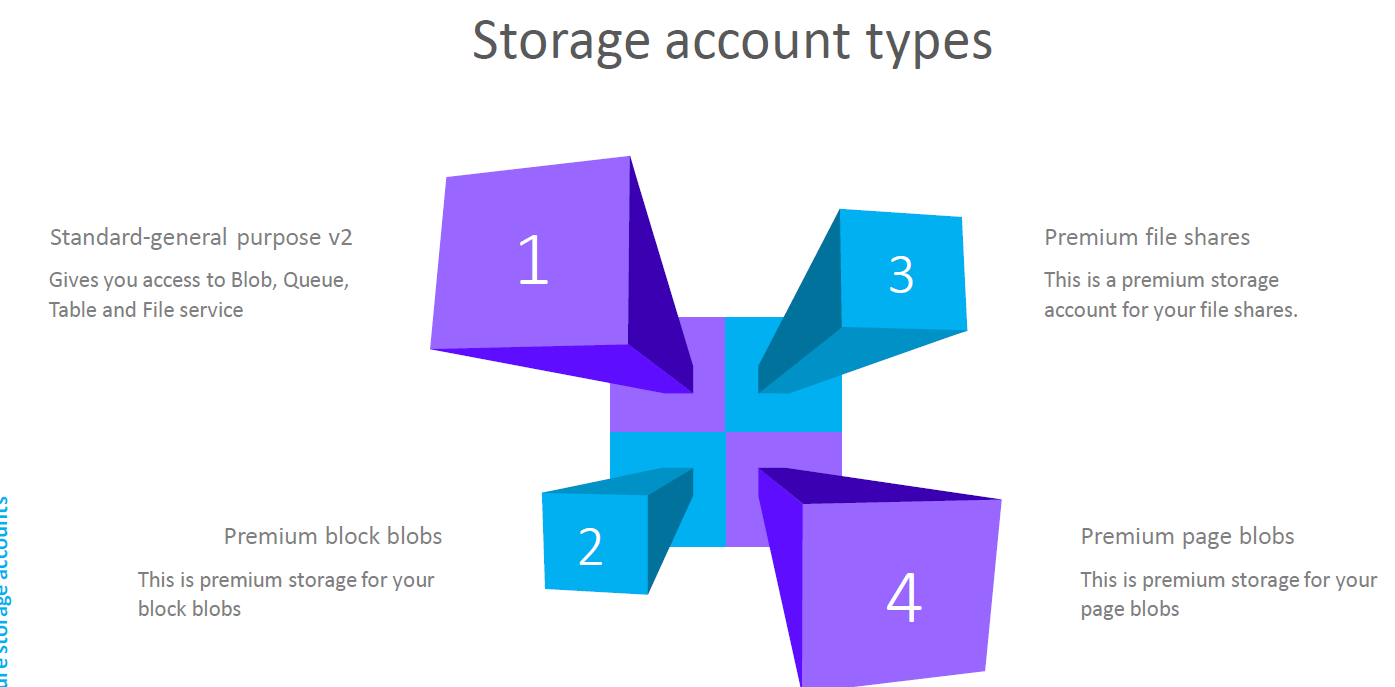
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Note : The general purposev2 storage acount we get call the services mentioned above  But for Premium Account type (Storage Account type ) – We can get a premium service for specific type like Block Service , File Service or Page blob Service   |  |  | | --- | --- | | BLOCK SERVICE | Its is used to store objects | | PAGE BLOB SERVICE | It is used to store hard disk drive of VM’s | |

### STORAGE ACCOUNT

* The general-purpose storage account is equipped with Container
* Containers are used for Storing BLOBs like images , video etc.)



### TYPES OF STORAGE ACCOUNT



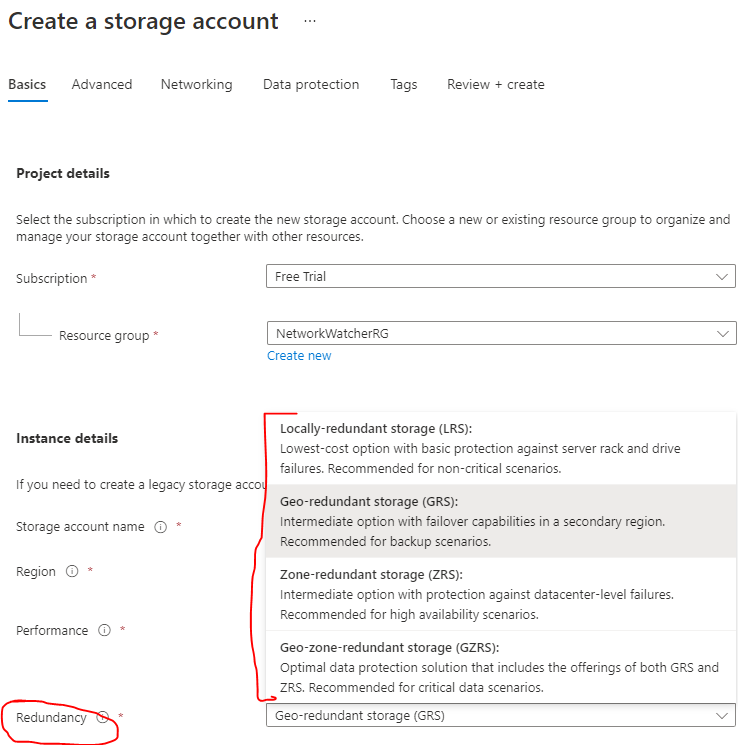
### DATA REDUNDANCY IN AZURE STORAGE ACCOUNT

#### WHAT IS REDUNDANCY?

|  |  |
| --- | --- |
|  | * When we store a data in a storage account – it is basically get stored in a physical storage device. * When azure store in the data in the storage devices – it stores multiple copies of the data * This helps in protecting the planned and unplanned events, transient hardware failures , network and power outage. |

The different statergies Azure uses to make multiple copies of the data are following

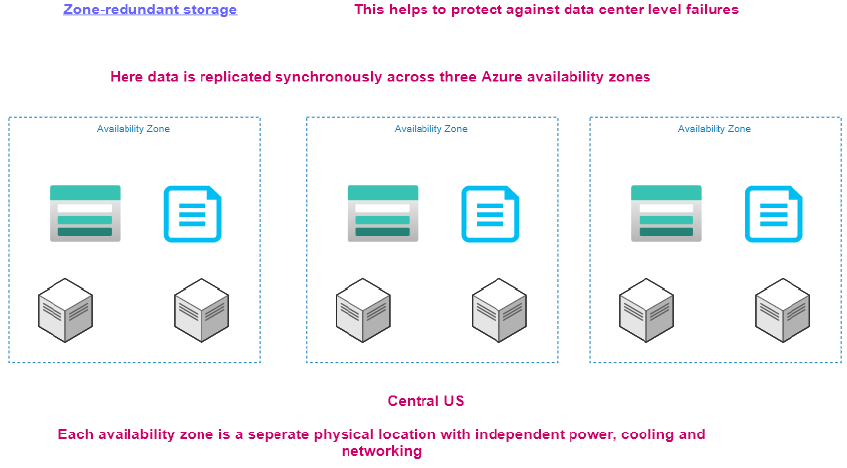
1. **LOCALLY-REDUNDANT STORAGE**
2. **GEO-REDUNDANT STORAGE**
3. **ZONE-REDUNDANT STORAGE**
4. **GEO-ZONE-REDUNDANT STORAGE**



#### LOCALLY-REDUNDANT STORAGE

|  |  |
| --- | --- |
|  | In this strategy of making the copies of data   * In an Azure data center 3 copies of data are made in different storage devices within a data center * Hence event if there is failure (like rack or device failure) in one of the storage devices – the data will be still available |

#### ZONE-REDUNDANT STORAGE



In Zone Redundant Storage

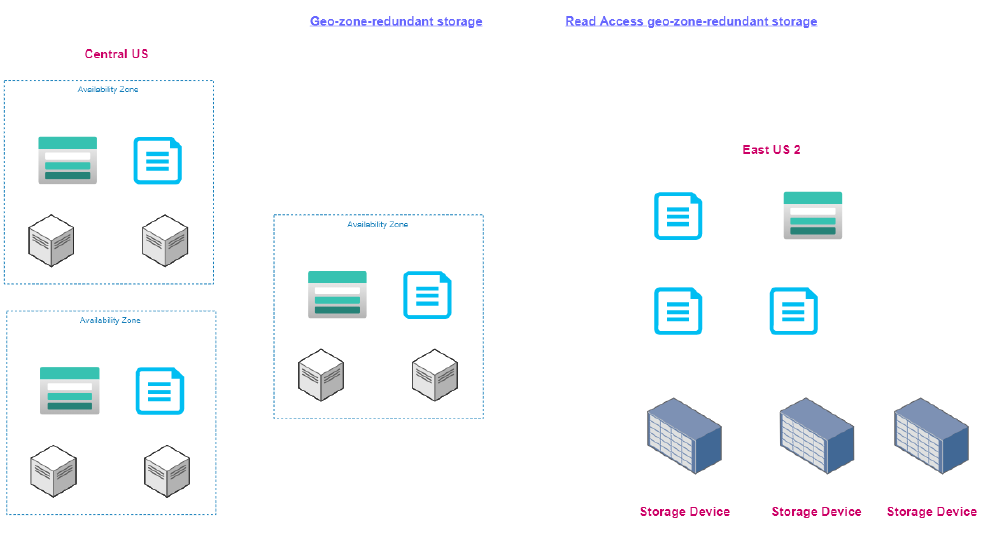
* This help in protecting against failure in a data center.
* Here the data is copied to 3 availability zones – Hence the data will be available even if a data center fails.

Note:

* Each availability zone resides in separate physical location.
* In this strategy- if the entire zone is not available (like Central US) , then all the availability zone will not be available. In this case we can go with Geo Redundant Storage.

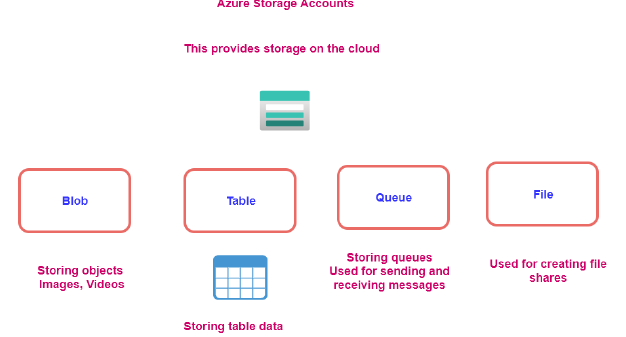
#### GEO-ZONE-REDUNDANT STORAGE

* The data is first replicated to availability zones in a particular zone.
* After that the data is replicated to secondary region



### CORE STORAGE SERVICES

|  |  |
| --- | --- |
| BLOB STORAGE | * Azure Blob Storage is an object storage solution that you can use to store massive amounts of unstructured data, such as text or binary data. * Blob Storage is ideal for serving images or documents directly to a browser, storing data for archivesor distributed access, streaming video and audio, and disaster recovery scenarios. |
| TABLE STORAGE | Azure Table Storage offers a NoSQL data store for key value pairs using large scale datasets.  You can use Azure Table Storage to store petabytes of semi-structured data,and keep your costs down. |
| QUEUE STORAGE | Azure Queue Storage provides asynchronous message queuing for communication between  application components,whether they're running in the cloud, on the desktop,on premises, or on mobile devices |
| FILE STORAGE | Azure File Storage offers fully managed file shares in the cloud, and shares are accessible  using industry standard network protocols.Mounting Azure file shares is just like connecting to shares on your local network. |
| DISK STORAGE | * Azure Disk Storage provides disks for virtual machines and applications to access and use as they need - similar to how they would access disks that were on premises. * Azure offers both solid state drives for higher performance workloads and conventional hard drives for your less critical business scenarios. |



#### DISK STORAGE

|  |  |
| --- | --- |
| The following illustration shows an Azure virtual machine that uses separate disks to store different data.Diagram of two disks inside a virtual machine. One stores the operating system and one stores data. | * Disk Storage provides disks for Azure virtual machines. Applications and other services can access and use these disks as needed, similar to how they would in on-premises scenarios. Disk Storage allows data to be persistently stored and accessed from an attached virtual hard disk. * Disks come in many different sizes and performance levels, from solid-state drives (SSDs) to traditional spinning hard disk drives (HDDs), with varying performance tiers. You can use standard SSD and HDD disks for less critical workloads, premium SSD disks for mission-critical production applications, and ultra disks for data-intensive workloads such as SAP HANA, top tier databases, and transaction-heavy workloads. |

#### BLOB STORAGE

|  |  |
| --- | --- |
| * Azure Blob Storage is an object storage solution for the cloud. It can store massive amounts of data, such as text or binary data. * Azure Blob Storage is unstructured, meaning that there are no restrictions on the kinds of data it can hold. * Blob Storage can manage thousands of simultaneous uploads, massive amounts of video data, constantly growing log files, and can be reached from anywhere with an internet connection. | Blb Storage is ideal for:   * Serving images or documents directly to a browser. * Storing files for distributed access. * Streaming video and audio. * Storing data for backup and restore, disaster recovery, and archiving. * Storing data for analysis by an on-premises or Azure-hosted service. * Storing up to 8 TB of data for virtual machines. |

##### CONTAINER

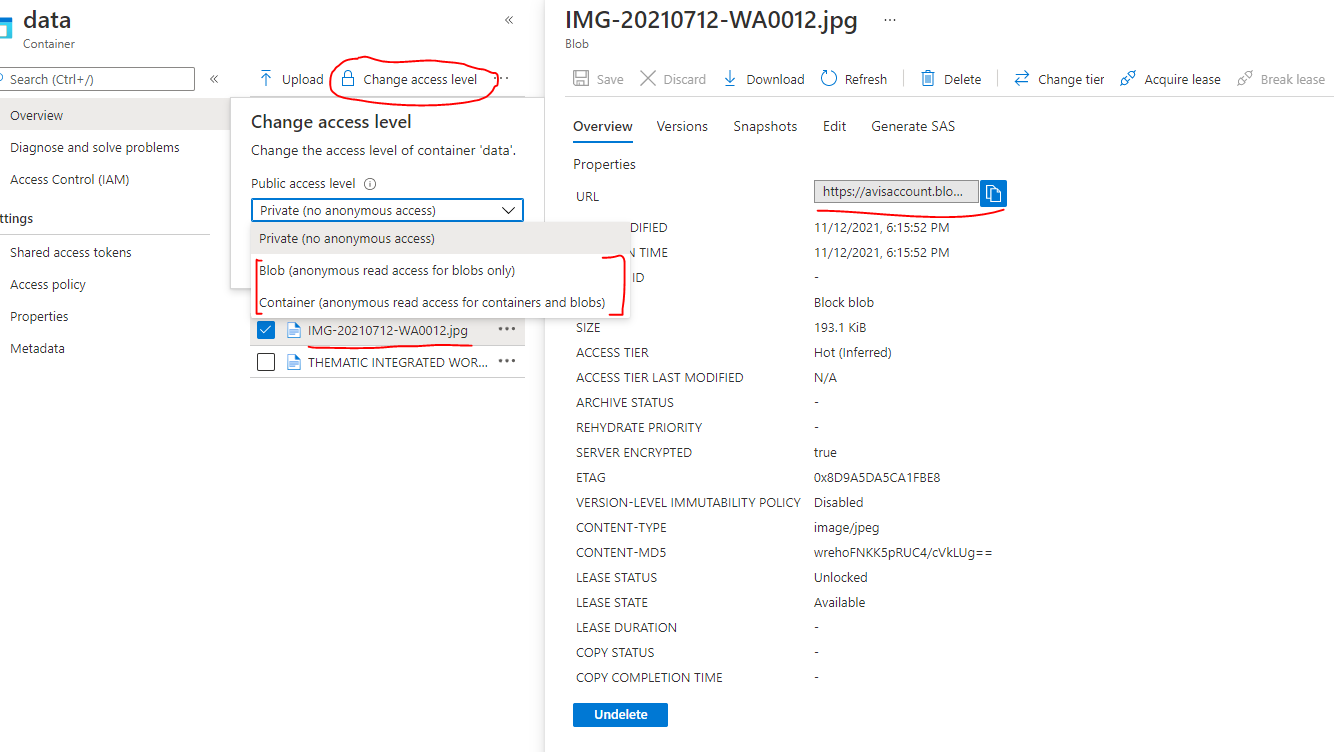
The containers helps as to organize the blobs.

###### CREATING A CONTAINER

|  |  |
| --- | --- |
|  |  |
| * Enter the name of the container. * The container will hold all the blobs like image and videos * Using the upload option, we can able to upload the images and videos * Note : ***Anything uploaded in a container will have unique url.*** | |

##### ACCESSING THE BLOB

* All the BLOB items have a unique url. Click on the items to view its property
* The URl field will give the get the unique url of the BLOB Item(e.g. : <https://avisaccount.blob.core.windows.net/data/IMG-20210712-WA0012.jpg> )
* ***The items will be accessible when the Access level is not private.***



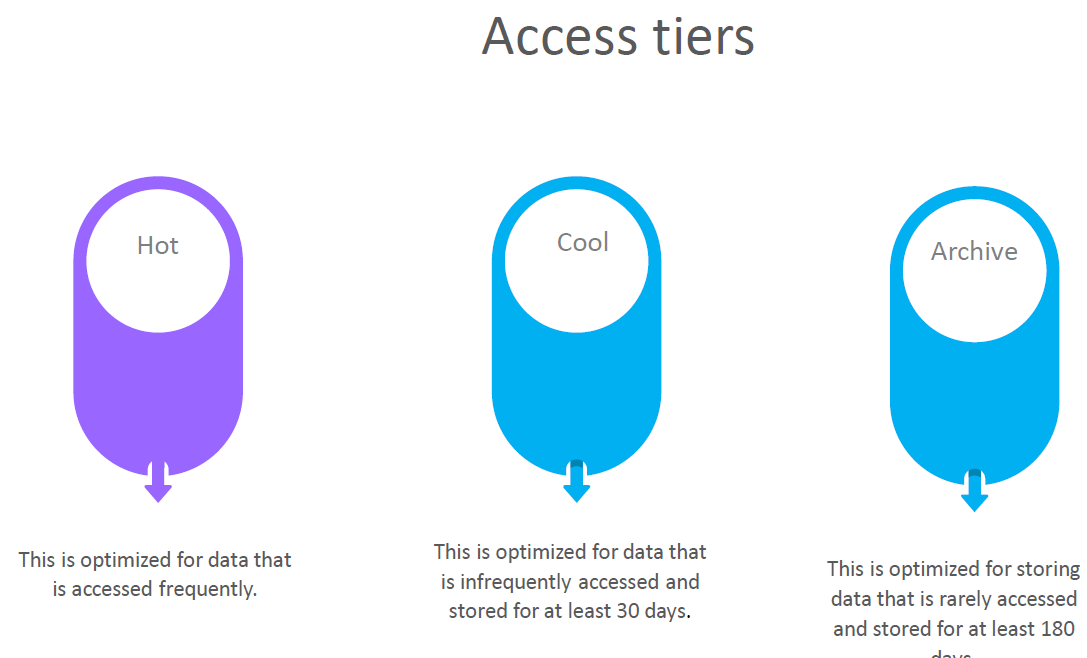
##### ACCESSING TIERS OF BLOBS



* Data stored in the cloud can grow at an exponential pace. To manage costs for expanding storage needs, it's helpful to organize your data based on attributes like ***frequency of access and planned retention period***.
* Data stored in the cloud can be different based on how it's generated, processed, and accessed over its lifetime. Some data is actively accessed and modified throughout its lifetime. Some data is accessed frequently early in its lifetime, with access dropping drastically as the data ages. Some data remains idle in the cloud and is rarely, if ever, accessed after it's stored. To accommodate these different access needs, Azure provides several access tiers, which you can use to balance your storage costs with your access needs.

***Azure Storage access tiers include***

Azure Storage offers different access tiers for your blob storage, helping you store object data in the most cost-effective manner. The available access tiers include:

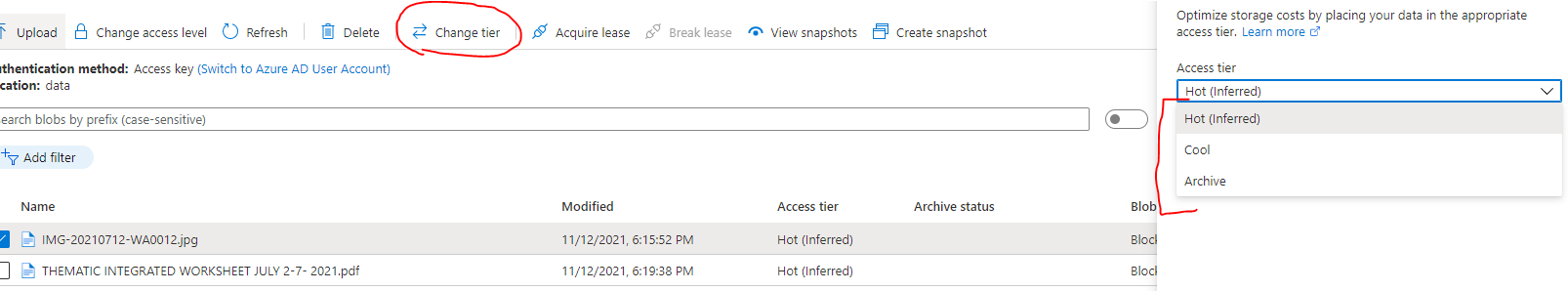


* **HOT ACCESS TIER**: Optimized for storing data that is accessed frequently (for example, images for your website).
* **COOL ACCESS TIER:** Optimized for data that is infrequently accessed and stored for at least 30 days (for example, invoices for your customers).
* **ARCHIVE ACCESS TIER**: Appropriate for data that is rarely accessed and stored for at least 180 days, with flexible latency requirements (for example, long-term backups).

The following considerations apply to the different access tiers:

* Only the hot and cool access tiers can be set at the account level. The archive access tier isn't available at the account level.
* Hot, cool, and archive tiers can be set at the blob level, during upload or after upload.
* Data in the cool access tier can tolerate slightly lower availability, but still requires high durability, retrieval latency, and throughput characteristics similar to hot data. For cool data, a slightly lower availability service-level agreement (SLA) and higher access costs compared to hot data are acceptable trade-offs for lower storage costs.
* Archive storage stores data offline and offers the lowest storage costs, but also the highest costs to rehydrate and access data.

The Access Tiers can be changed as shown below. Select the BLOB item 🡪 Change Tier 🡪 Select the Access Tier.



#### FILE STORAGE

Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block and Network File System (preview) protocols. Azure file shares can be mounted concurrently by cloud or on-premises deployments of Windows, Linux, and macOS. Applications running in Azure virtual machines or cloud services can mount a file storage share to access file data, just as a desktop application would mount a typical SMB share. Any number of Azure virtual machines or roles can mount and access the file storage share simultaneously. Typical usage scenarios would be to share files anywhere in the world, diagnostic data, or application data sharing.

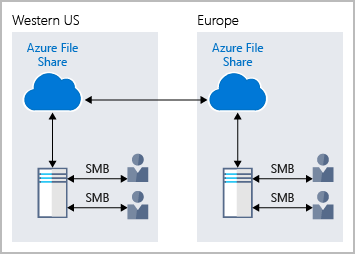
Use Azure Files for the following situations:

Many on-premises applications use file shares. Azure Files makes it easier to migrate those applications that share data to Azure. If you mount the Azure file share to the same drive letter that the on-premises application uses, the part of your application that accesses the file share should work with minimal changes, if any.

Store configuration files on a file share and access them from multiple VMs. Tools and utilities used by multiple developers in a group can be stored on a file share, ensuring that everybody can find them, and that they use the same version.

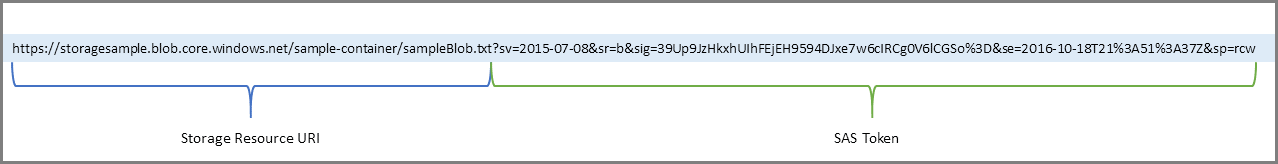
Write data to a file share, and process or analyze the data later. For example, you might want to do this with diagnostic logs, metrics, and crash dumps.

The following illustration shows Azure Files being used to share data between two geographical locations. Azure Files ensures the data is encrypted at rest, and the SMB protocol ensures the data is encrypted in transit.



One thing that distinguishes Azure Files from files on a corporate file share is that you can access the files from anywhere in the world, by using a URL that points to the file. You can also use Shared Access Signature (SAS) tokens to allow access to a private asset for a specific amount of time.

Here's an example of a service SAS URI, showing the resource URI and the SAS token:



#### TABLE

* Azure Table storage is a service that stores **non-relational structured data (also known as structured NoSQL data)** in the cloud, providing a key/attribute store with a schemeless design.
* Because Table storage is schemeless,
  + it's easy to adapt your data as the needs of your application evolve.
  + Access to Table storage data is fast and cost-effective for many types of applications and is typically lower in cost than traditional SQL for similar volumes of data.
* We can use Table storage to store flexible datasets like user data for web applications, address books, device information, or other types of metadata your service requires.
* **We can store any number of entities in a table, and a storage account may contain any number of tables, up to the capacity limit of the storage account**.

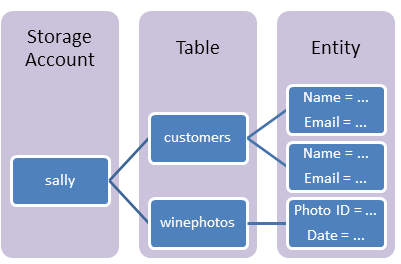
##### WHAT IS TABLE STORAGE

* Azure Table storage stores large amounts of structured data. The service is a NoSQL datastore which accepts authenticated calls from inside and outside the Azure cloud. Azure tables are ideal for storing structured, non-relational data.

*Common uses of Table storage include*

* Storing TBs of structured data capable of serving web scale applications
* Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
* Quickly querying data using a clustered index
* Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries
* We can use Table storage to store and query huge sets of structured, non-relational data, and the tables will scale as demand increases.

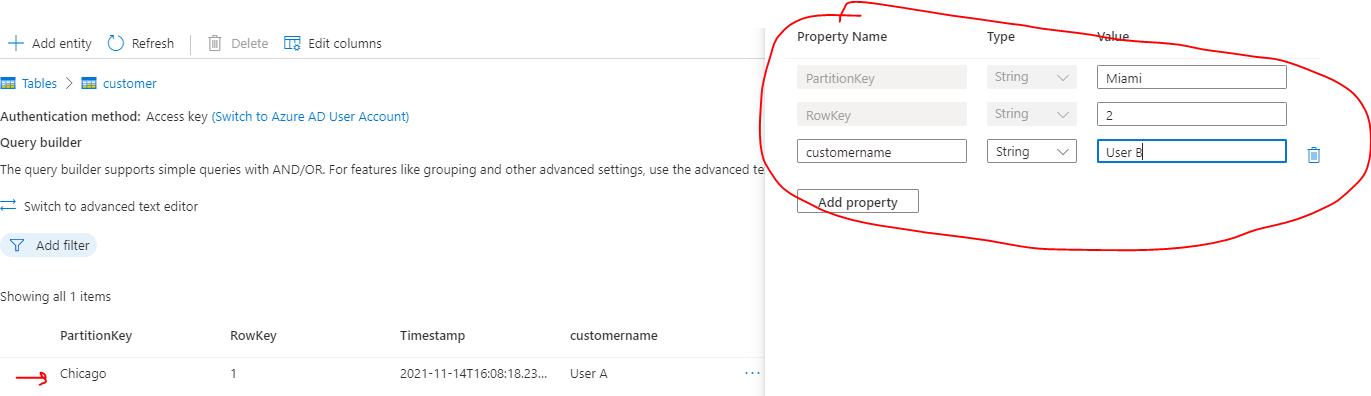
##### TABLE STORAGE COMPONENTS



|  |  |
| --- | --- |
| **URL FORMAT** | * Azure Table Storage accounts format: http://<storage account>.table.core.windows.net/<table> * Azure Cosmos DB Table API accounts format: http://<storage account>.table.cosmosdb.azure.com/<table> |
| **TABLE** | A table is a collection of entities. Tables don't enforce a schema on entities, which means a single table can contain entities that have different sets of properties. |
| **ENTITY** | An entity is a set of properties, like a database row. **An entity in Azure Storage can be up to 1MB in size. An entity in Azure Cosmos DB can be up to 2MB in size** |
| **PROPERTIES** | * A property is a name-value pair. * Each entity can include up to 252 properties to store data. * Each entity also has three system properties that specify a partition key, a row key, and a timestamp. **Entities with the same partition key can be queried more quickly and inserted/updated in atomic operations.** * An entity's row key is its unique identifier within a partition. |

##### ADDING DATA TO TABLE

|  |  |
| --- | --- |
|  | * For table storage – we can create tables in the azure storage account. * To add data to the table 🡪 Navigate to Storage Browser 🡪 Navigate to the table 🡪 Add Entity * An Entity has 2 parts   + Partition key   + RowKey |



###### PARTITION KEY AND ROWKEY

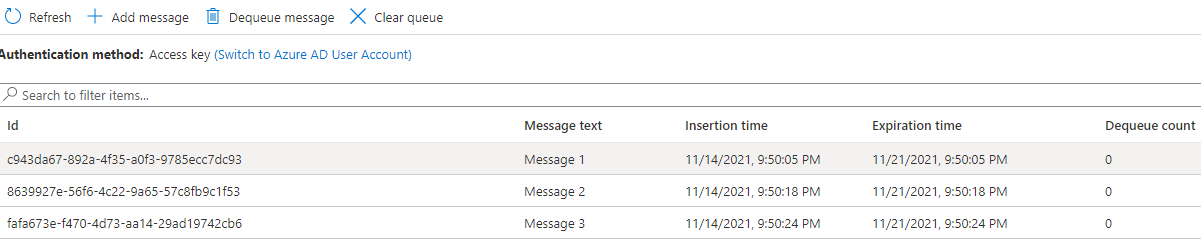
* PARTITION KEY: If we have huge amount of data in a table, dividing the table into partition – it becomes easier to search for an entity in particular partition.
* ROWKEY: Rowkey helps in searching the data within the partition.

Example: Let’s say we are creating a “customer” table. The table can be divided in partition, it can be customer’s city. The partitioning helps fast searching within the table. Rowkey on the other hand can the customer Id which can help in searching within the partition.

#### QUEUES

|  |  |
| --- | --- |
|  |  |

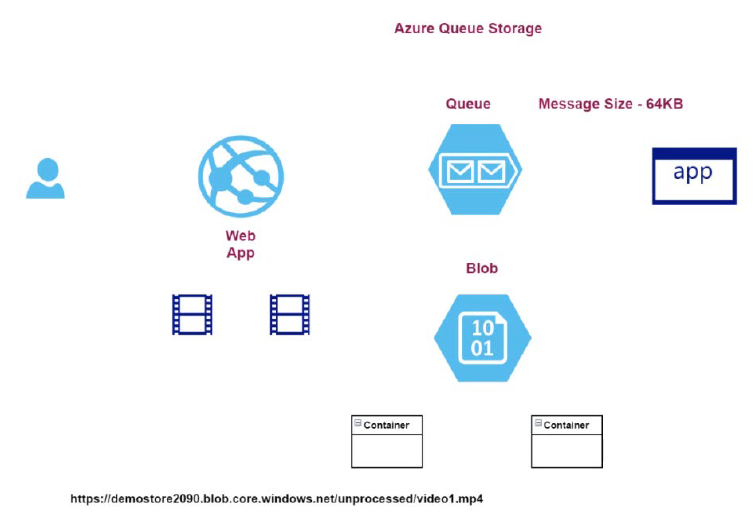
* Dequeue message will remove the message from top of the queue
* Clear queue will remove all the message of the queue.



Note in the queue

* The messages are added / dequeued from the queue programmatically.

#### EXAMPLE

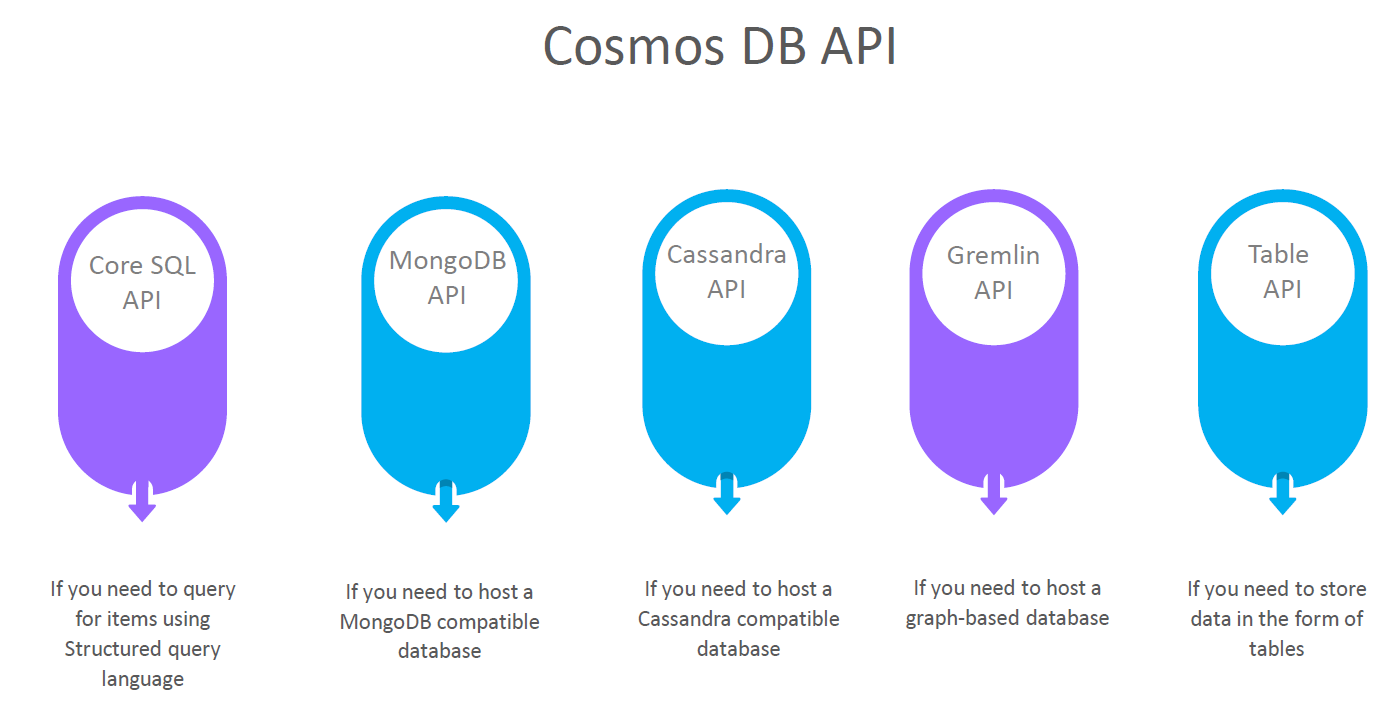


### AZURE COSMOS DATABASE

* This is a fully managed NoSQL database.
* The database provides fast response time and is highly scalable.
* Here the underlying infrastructure is completely managed by Azure.
* Commonly used for web, mobile, gaming and IoT applications that need to handle massive amounts of data.
* Azure Cosmos DB is a globally distributed, multi-model database service.
* You can elastically and independently scale throughput and storage across any number of Azure regions worldwide.
* Azure Cosmos DB supports schemaless data, which lets you build highly responsive and "Always On" applications to support constantly changing data.
* You can use this feature to store data that's updated and maintained by users around the world.

|  |  |
| --- | --- |
| **COSMOS DB** | **SQL DB** |
| We need to have relationship between tables | No SQL data store |
| When we you want to have constraints like foreign key constraint | Flexible Schema  No need of joins between data structures. |

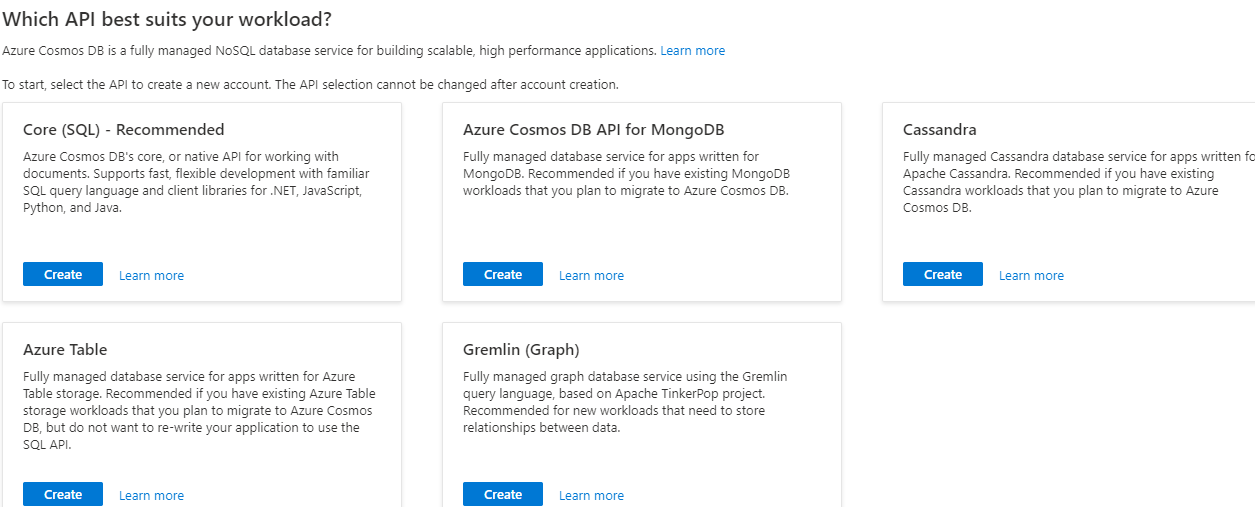
#### COSMOS DB API



### CREATING A COSMOS DB ACCOUNT

#### SELECTING COSMOS DB API

1. We want to host a Data where we can issue SQL based command 🡪 Select Core(SQL) API.
2. All the data is stored in form of JSON when it comes to underlying document.



#### COSMOS DB ACCOUNT CONFIGURATION

|  |  |
| --- | --- |
|  | 1. When it comes to costing of the cosmos Db account it can be    1. Provisioned throughput    2. Serverless   PROVISION THROUGHOUT:  In this case, we are not charged separately for CPU , Memory etc. but we are charged based on RU(request unit) which is kind bundle of CPU , memory etc.. |

#### CREATING A COSMOS DATABASE AND CONTAINER

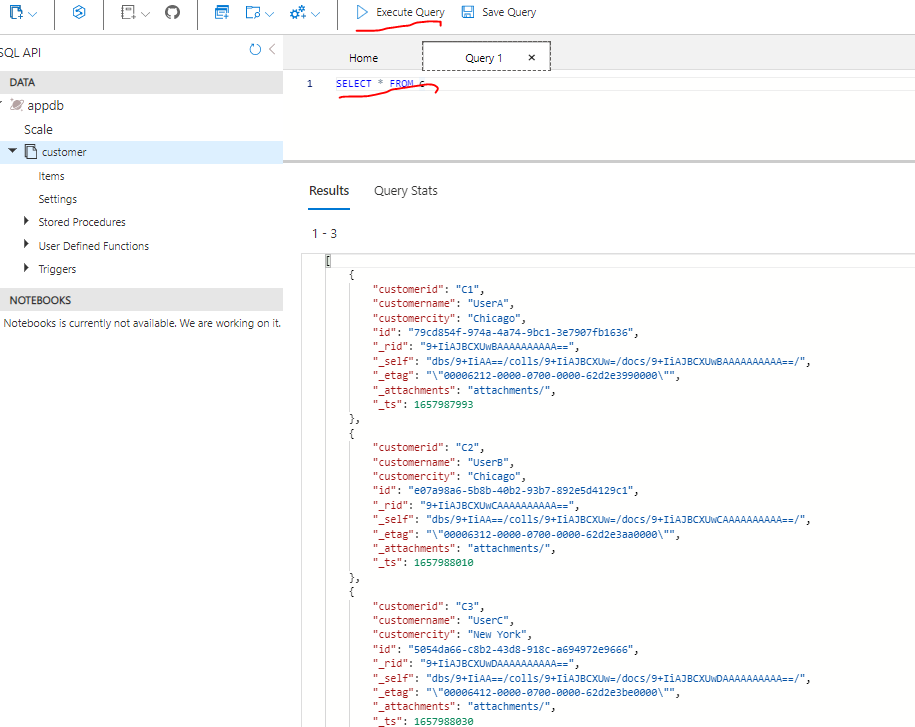
* Go to Data explorer 🡪 New Container

|  |  |
| --- | --- |
|  | * Step 1: We need to create a database * Step 2: The database contains container. Each database can have multiple containers. Container is equivalent to table in relational DB * Step 3: Container can have items. Item is equivalent to one record in relational DB   **CREATED DB AND CONTAINER** |

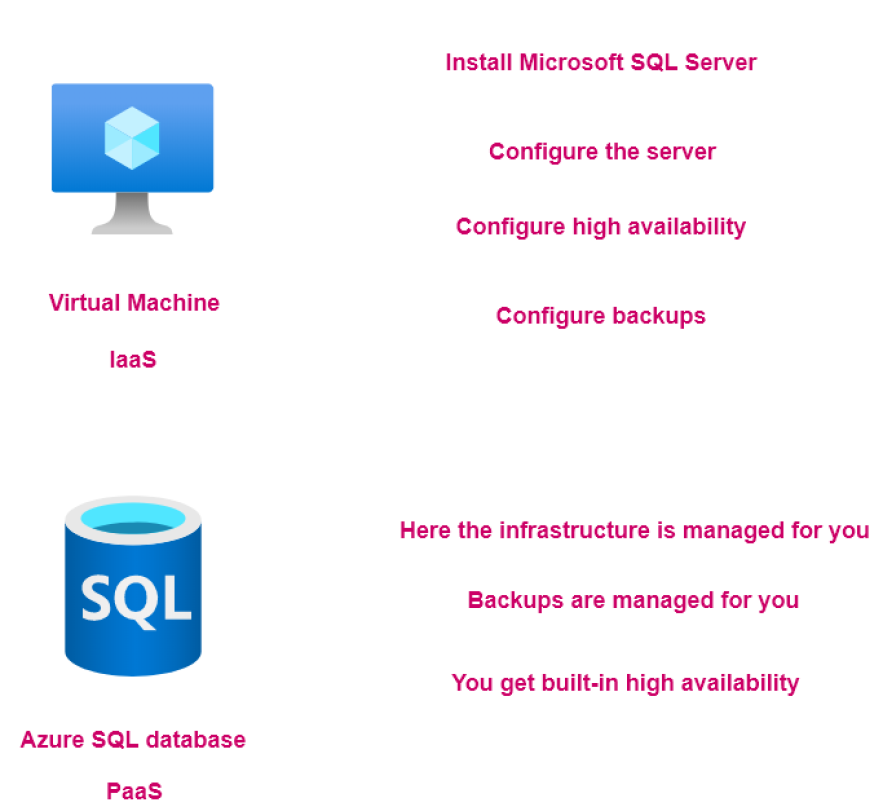
#### CREATING A COSMOS DATABASE AND CONTAINER

|  |  |
| --- | --- |
|  | * To add new items in the container 🡪 New Items   DATA FORMAT  {  "customerid":"C1",  "customername":"UserA",  "customercity":"Chicago"  } |

#### QUERYING CONTAINER DATA



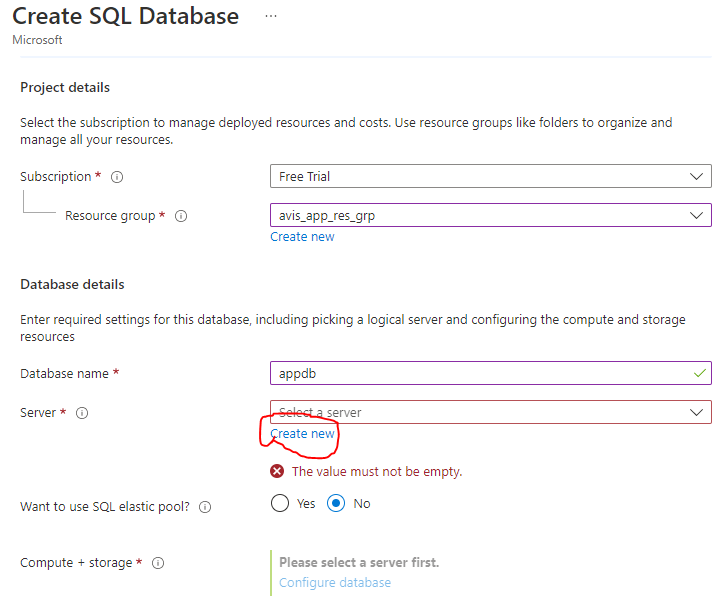
### AZURE SQL DATABASE



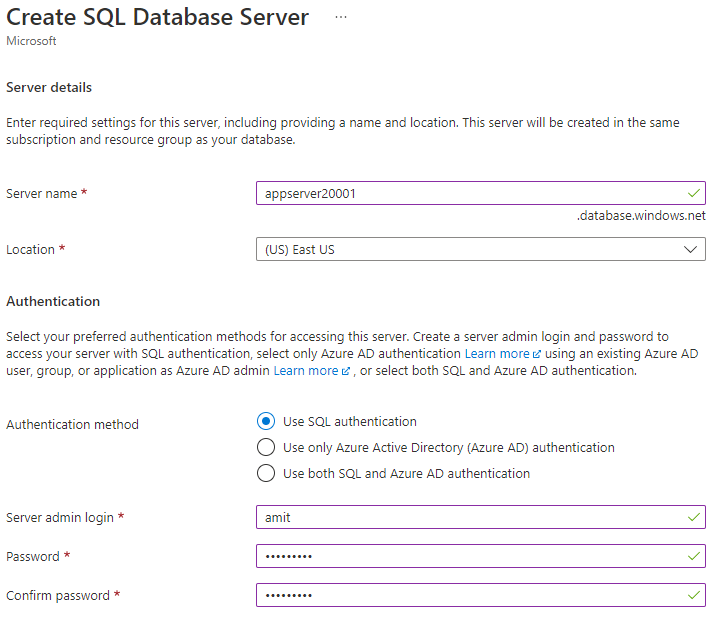
#### CREATING A SQL DATABASE SERVER

* When we use SQL Database service – it creates two resources
  + Azure SQL Database Server : This is managed by Azure itself. That’s the Azure SQL database is known has PaaS (Platform as a Service)
  + Azure SQL Database: To host the tables of data

STEP 1: CREATE THE SQL DATABASE



STEP 2: CREATE THE SQL DATABASE SERVER



STEP 3: CONFIGURING THE SQL DATABASE (COMPUTE AND STOARAGE CONFIGURATION)

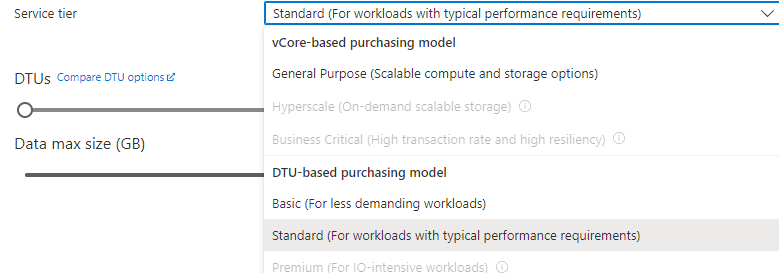
SQL database service pricing tier are classified in two broad categories:

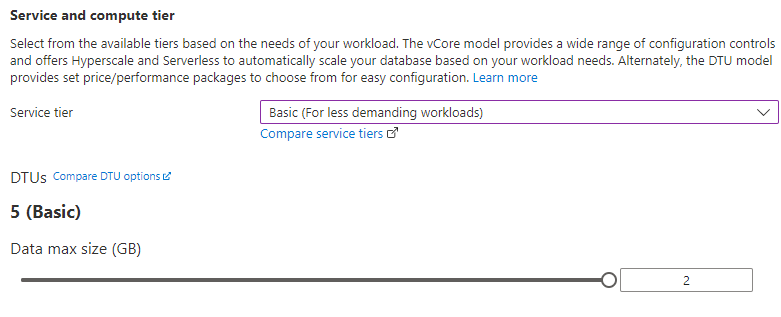
**DTU –DATABASE TRANSACTION UNITS.**

* This is a blended measure of CPU, Memory and Input/Output.
* There are different pricing tiers when it comes to the DTU model.

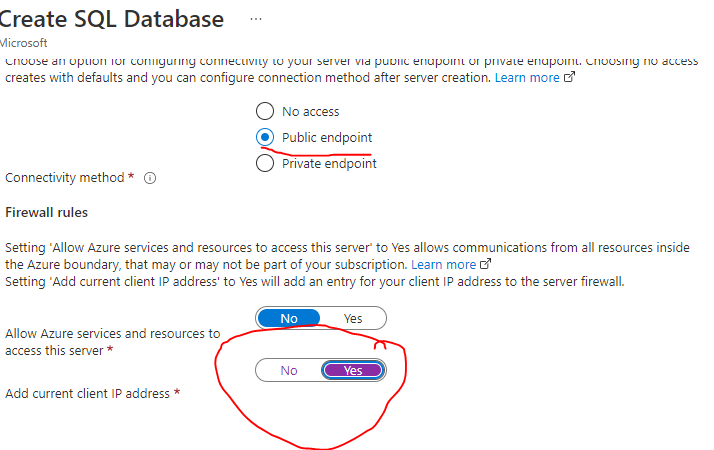
**VCORE-BASED PURCHASING MODEL.**

* Here you can independently scale compute and storage.
* You can make use of the hybrid benefit model. Here you can save on costs if you have existing SQL Server licenses.





NETWORKING

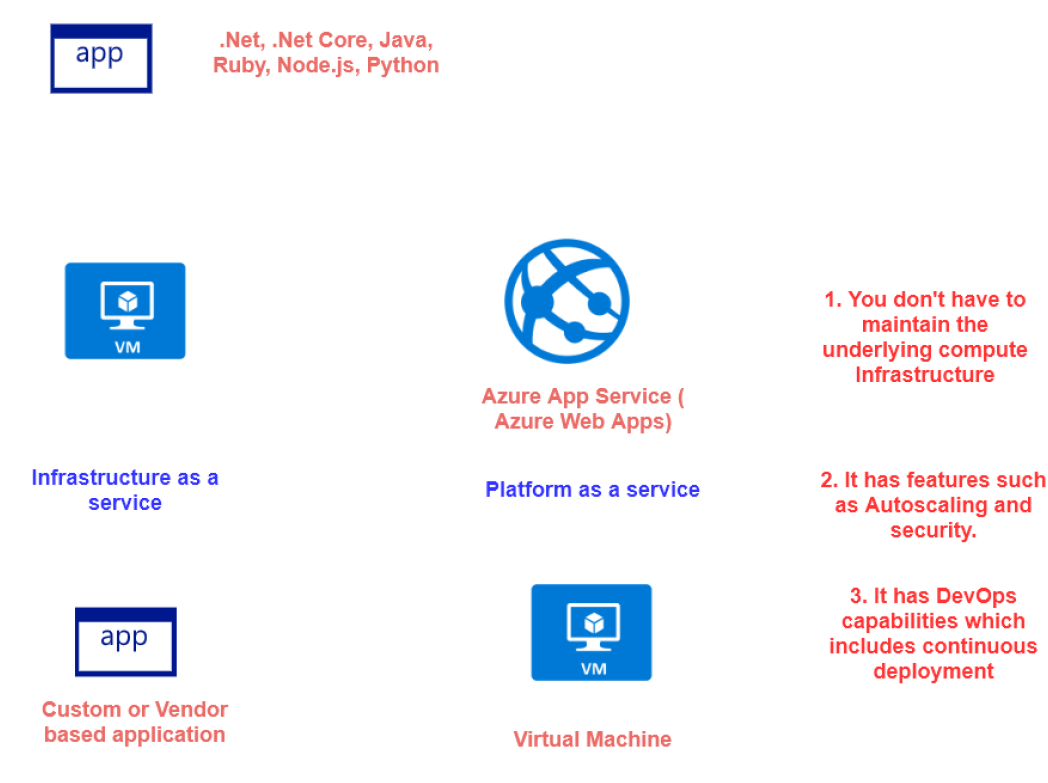


## AZURE WEB-APP SERVICE

* This is an HTTP-based service used for hosting web applications.
* The applications can be in .NET, .NET Core, Java, Ruby, Node.js or Python.
* Applications can run both on Windows or Linux-based platforms.
* This is a platform-as-a-service where the infrastructure is managed for us. Bu azure

***Note: These apps can also be hosted on VM, but this will be infrastructure as service. Here we must maintain the underlying VM on our own. For example – to host node-based app – it will be our job to install Node on the VM. We use this strategy when we need to deploy vendor-based application like AEM***

* Now in this platform has a service, the application gets deployed on an underlying virtual machine or an underlying compute infrastructure. Now, this entire infrastructure is managed by Azure themselves i.e., by the Azure App Service. Here we don't have to manage the underlying infrastructure like the underlying web server or underlying configuration of that web server. All of this is matched by the Azure App Service itself.

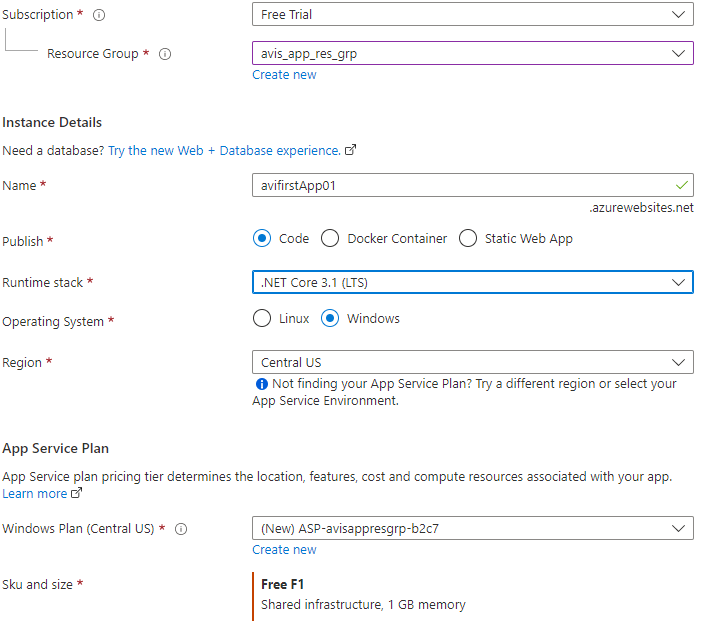


### ADVANTAGE OF USING APP SERVICE

* We don’t have to manage the underlying compute infrastructure.
* Features such as auto scaling and security,
* It also has DevOps capabilities.

### CREATING AN APP SERVICE

* Select “web App” service from the Create Resource

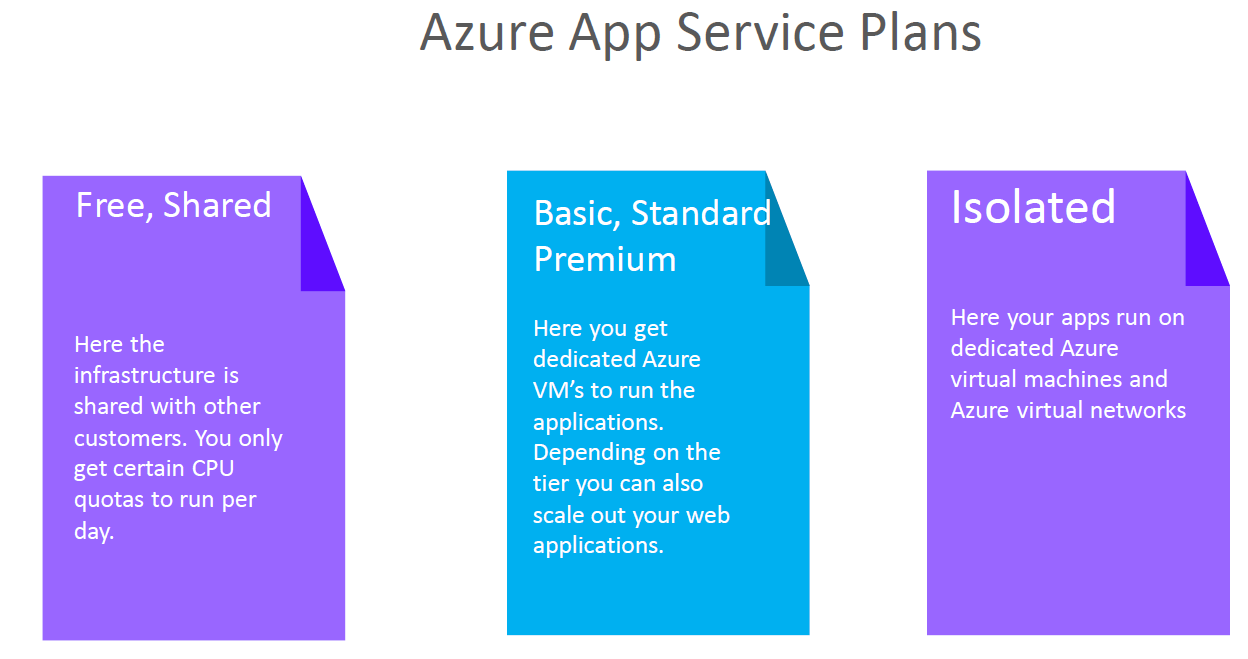


|  |  |
| --- | --- |
| Name | Unique name of the app. By default, azure append a default DNS name to the web app. We can have custom DNS name as well. |
| Publish | Type of App |
| Runtime Stack | Runtime of the of application is going to be hosted |

Once the setup is done. Azure will host a default “.NET Core” based application on Azure webapp service . For our case it can be accessed at - <https://avifirstapp01.azurewebsites.net/>

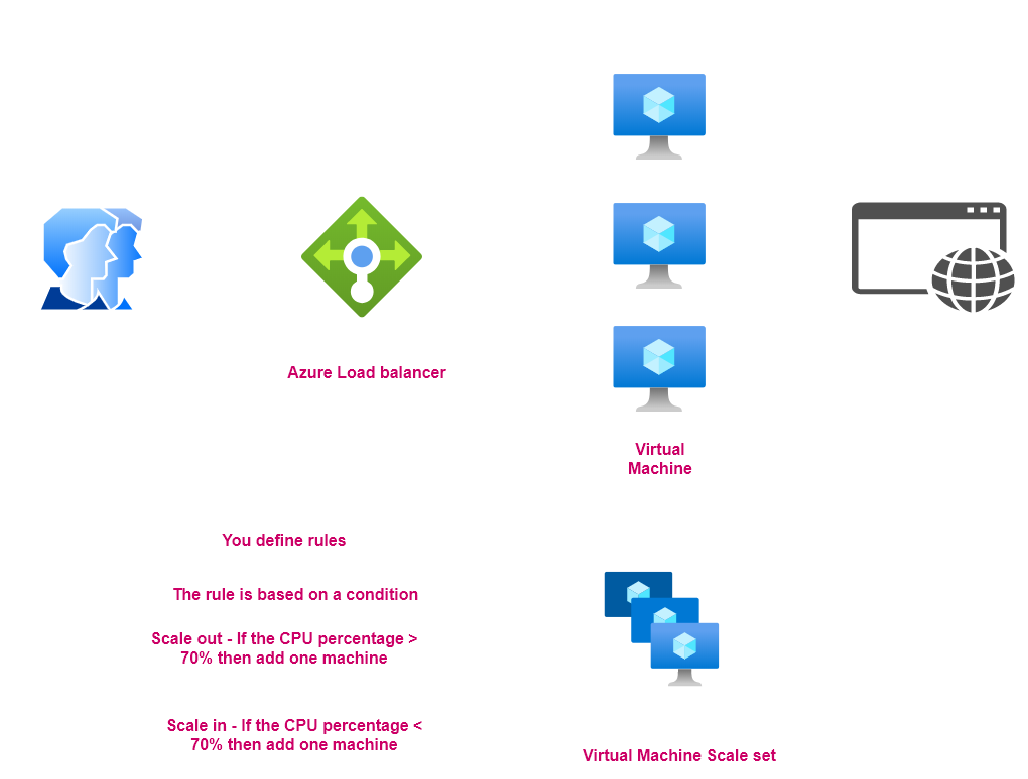
### AZURE APP SERVICE PLAN

* App service plan defines the set of compute resources that are used to run the web application

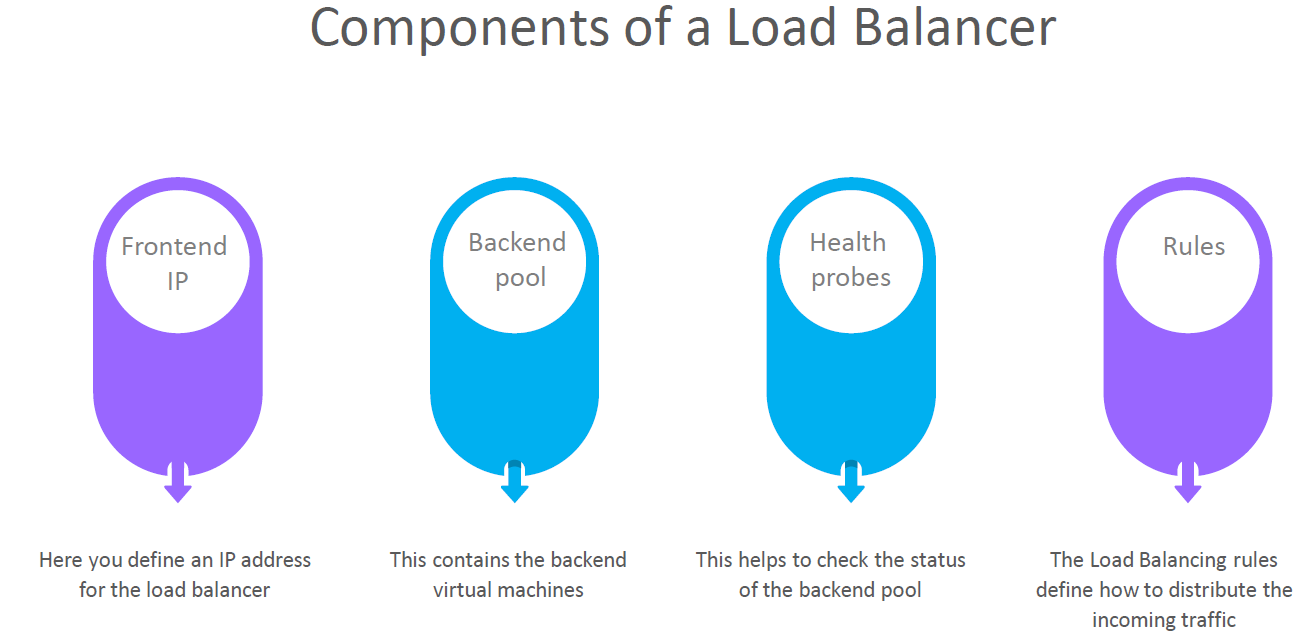


## AZURE LOAD BALANCER

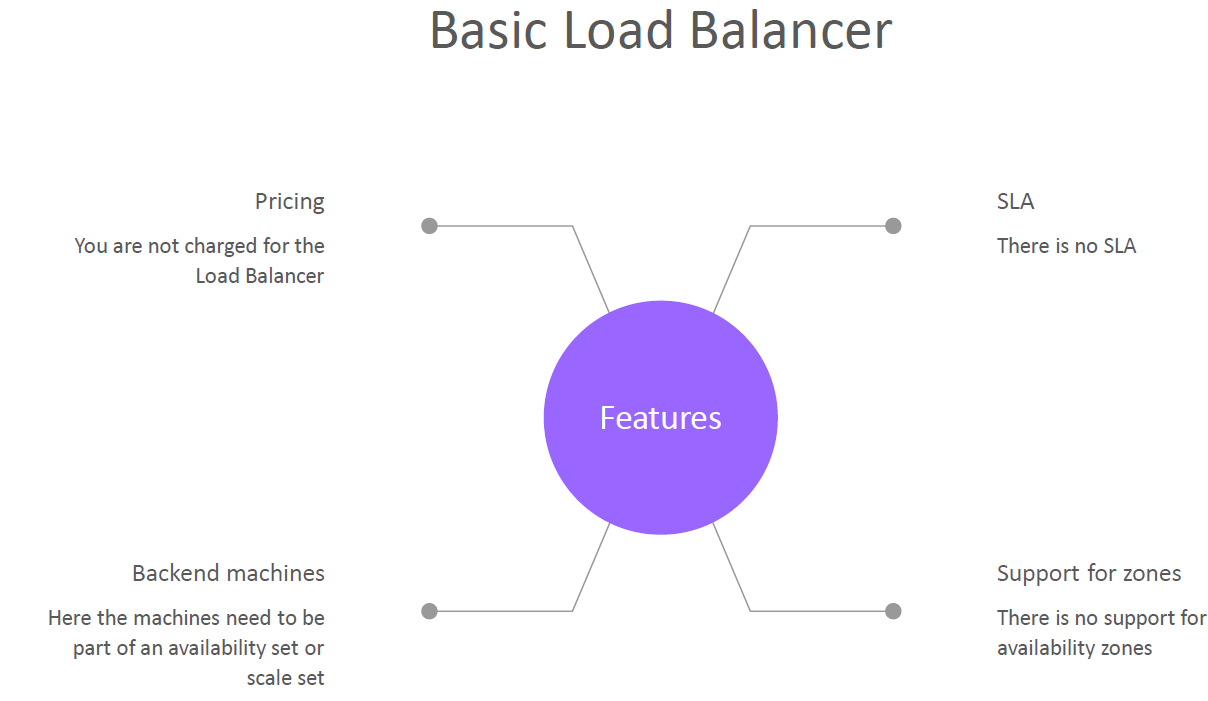
* This service is used to distribute the incoming network traffic across a group of backend resources of servers
* You can define two types of load balancers –***Public or Private Load Balancers***
* You have 2 SKUs for the Load Balancer –Standard and Basic Load Balancer



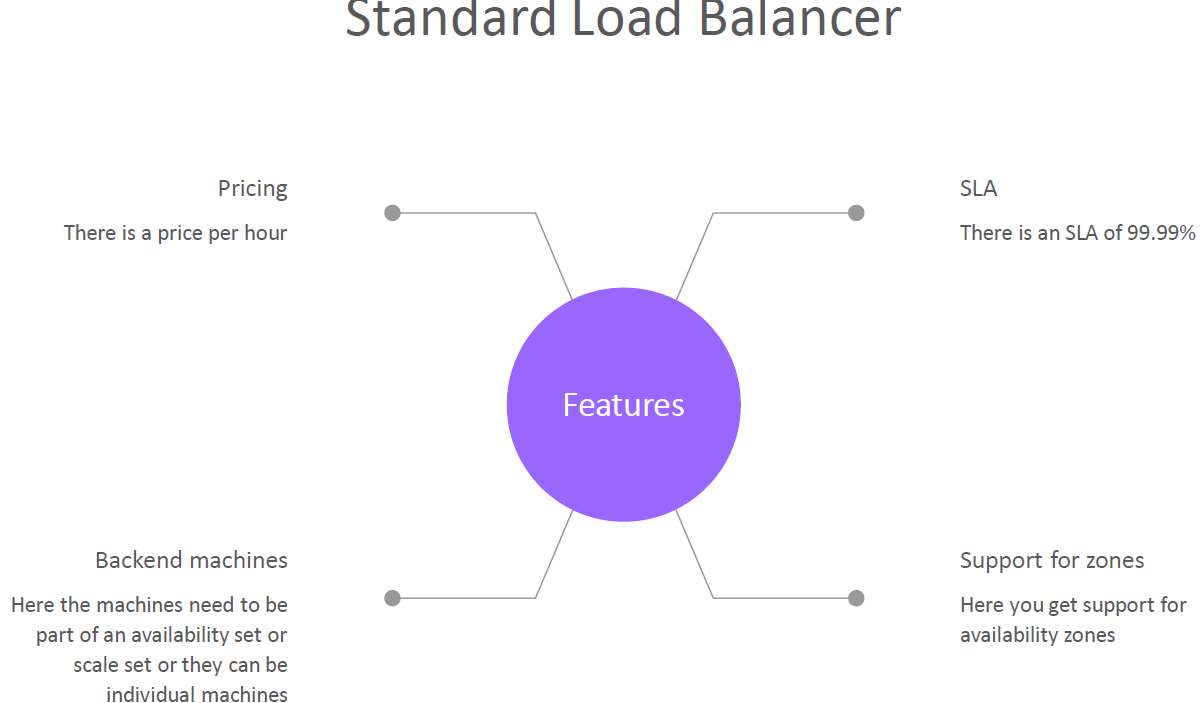
### COMPONENTS OF LOAD BALANCER



### BASIC LOAD BALANCER



### STANDARD LOAD BALANCER



## SERVERLESS COMPUTING IN AZURE

* We go for the serverless features of Azure our application logic is event driven. In other words, for a large amount of time, our application is waiting for a particular input before it performs any processing. To reduce your costs, you want to avoid having to pay for the time that your application is waiting for input.

**Serverless** computing is the abstraction of servers, infrastructure, and operating systems. With serverless computing, Azure takes care of managing the server infrastructure and the allocation and deallocation of resources based on demand. Infrastructure isn't your responsibility. Scaling and performance are handled automatically. You're billed only for the exact resources you use. There's no need to even reserve capacity.

*Serverless computing includes the abstraction of servers, an event-driven scale, and micro-billing:*

* **Abstraction of servers**: Serverless computing abstracts the servers you run on. You never explicitly reserve server instances. The platform manages that for you. Each function execution can run on a different compute instance. This execution context is transparent to the code. With serverless architecture, you deploy your code, which then runs with high availability.
* **Event-driven scale**: Serverless computing is an excellent fit for workloads that respond to incoming events. Events include triggers by:
  + Timers, for example, if a function needs to run every day at 10:00 AM UTC.
  + HTTP, for example, API and webhook scenarios.
  + Queues, for example, with order processing.

Instead of writing an entire application, the developer authors a function, which contains both code and metadata about its triggers and bindings. The platform automatically schedules the function to run and scales the number of compute instances based on the rate of incoming events. Triggers define how a function is invoked. Bindings provide a declarative way to connect to services from within the code.

* **Micro-billing**: Traditional computing bills for a block of time like paying a monthly or annual rate for website hosting. This method of billing is convenient but isn't always cost effective. Even if a customer's website gets only one hit a day, they still pay for a full day's worth of availability. With serverless computing, they pay only for the time their code runs. If no active function executions occur, they're not charged. For example, if the code runs once a day for two minutes, they're charged for one execution and two minutes of computing time.

Azure has two implementations of serverless compute:

* **AZURE FUNCTIONS**: Functions can execute code in almost any modern language.
* **AZURE LOGIC APPS**: Logic apps are designed in a web-based designer and can execute logic triggered by Azure services without writing any code.

### AZURE FUNCTIONS

When you're concerned only about the code running your service, and not the underlying platform or infrastructure, using Azure Functions is ideal. Functions are commonly used when you need to perform work in response to an event (often via a REST request), timer, or message from another Azure service, and when that work can be completed quickly, within seconds or less.

Functions scale automatically based on demand, so they're a solid choice when demand is variable. For example, you might receive messages from an IoT solution that's used to monitor a fleet of delivery vehicles. You'll likely have more data arriving during business hours.

Using a virtual machine-based approach, you'd incur costs even when the virtual machine is idle. With functions, Azure runs your code when it's triggered and automatically deallocates resources when the function is finished. In this model, you're only charged for the CPU time used while your function runs.

Functions can be either stateless or stateful. When they're stateless (the default), they behave as if they're restarted every time they respond to an event. When they're stateful (called Durable Functions), a context is passed through the function to track prior activity.

Functions are a key component of serverless computing. They're also a general compute platform for running any type of code. If the needs of the developer's app change, you can deploy the project in an environment that isn't serverless. This flexibility allows you to manage scaling, run on virtual networks, and even completely isolate the functions.

### AZURE LOGIC APPS

Logic apps are similar to functions. Both enable you to trigger logic based on an event. Where functions execute code, logic apps execute workflows that are designed to automate business scenarios and are built from predefined logic blocks.

Every Azure logic app workflow starts with a trigger, which fires when a specific event happens or when newly available data meets specific criteria. Many triggers include basic scheduling capabilities, so developers can specify how regularly their workloads will run. Each time the trigger fires, the Logic Apps engine creates a logic app instance that runs the actions in the workflow. These actions can also include data conversions and flow controls, such as conditional statements, switch statements, loops, and branching.

You create logic app workflows by using a visual designer on the Azure portal or in Visual Studio. The workflows are persisted as a JSON file with a known workflow schema.

Azure provides more than 200 different connectors and processing blocks to interact with different services. These resources include the most popular enterprise apps. You can also build custom connectors and workflow steps if the service you need to interact with isn't covered. You then use the visual designer to link connectors and blocks together. You pass data through the workflow to do custom processing, often all without writing any code.

As an example, let's say a ticket arrives in Zendesk. You could:

Detect the intent of the message with cognitive services.

Create an item in SharePoint to track the issue.

Add the customer to your Dynamics 365 CRM system if they aren't already in your database.

Send a follow-up email to acknowledge their request.

All of those actions could be designed in a visual designer, which makes it easy to see the logic flow. For this reason, it's ideal for a business analyst role.

### FUNCTIONS VS. LOGIC APPS

### 