Contents

[CLOUD COMPUTING 3](#_Toc124620569)

[OSI MODEL 3](#_Toc124620570)

[SERVER TYPES 3](#_Toc124620571)

[BACKGROUND OF CLOUD COMPUTING 3](#_Toc124620572)

[ISSUES WITH IN-PREMISE INFRASTRUCTURE 3](#_Toc124620573)

[COST TYPES - CAPITAL EXPENSES VS. OPERATING EXPENSES 4](#_Toc124620574)

[SOLUTION OF ISSUES OF IN-PREMISE INFRASTRUCTURE 4](#_Toc124620575)

[VIRTUALIZATION 4](#_Toc124620576)

[VIRTUALIZED ENVIRONMENT 5](#_Toc124620577)

[BENEFITS OF VIRTUALIZATION 5](#_Toc124620578)

[TYPES OF HYPERVISORS 5](#_Toc124620579)

[CLOUD CONCEPTS 6](#_Toc124620580)

[WHAT IS CLOUD COMPUTING 6](#_Toc124620581)

[BENEFITTS OF CLOUD 6](#_Toc124620582)

[TYPES OF CLOUD MODELS 7](#_Toc124620583)

[SERVICE MODELS 7](#_Toc124620584)

[DEPLOYMENT MODELS 9](#_Toc124620585)

[WHAT IS SERVERLESS COMPUTING? 10](#_Toc124620586)

[AZURE 10](#_Toc124620587)

[HOW AZURE WORKS? 10](#_Toc124620588)

[AZURE INFRASTRUCTURE SET UP 10](#_Toc124620589)

[AZURE SUBSCRIPTIONS, MANAGEMENT GROUPS, AND RESOURCES 10](#_Toc124620590)

[WORKLOAD 11](#_Toc124620591)

[RESOURCE GROUP 11](#_Toc124620592)

[MOVING THE RESOURCES ACROSS RESOURCE GROUP 12](#_Toc124620593)

[MOVING THE RESOURCES ACROSS SUBSCRIPTION 12](#_Toc124620594)

[RESOURCE TAGS 13](#_Toc124620595)

[RESOURCE LOCKING 14](#_Toc124620596)

[AZURE RESOURCE MANAGER (ARM) 15](#_Toc124620597)

[ACCESSING AZURE PLATFORM 16](#_Toc124620598)

[POWERSHELL 16](#_Toc124620599)

[AZURE CLI 20](#_Toc124620600)

[AZURE MARKET PLACE 21](#_Toc124620601)

[AZURE REGIONS AND ZONES 22](#_Toc124620602)

[WHY REGION IS IMPORTANT 22](#_Toc124620603)

[SELECTING A REGION 23](#_Toc124620604)

[AZURE AVAILABILITY ZONE 23](#_Toc124620605)

[AZURE REGION PAIR 24](#_Toc124620606)

[AZURE SERVICES 24](#_Toc124620607)

[COMPUTE SERVICES 24](#_Toc124620608)

[NETWORKING 25](#_Toc124620609)

[STORAGE SERVICE 27](#_Toc124620610)

[DATABASE 27](#_Toc124620611)

[AZURE CORE SERVICES 28](#_Toc124620612)

[AZURE COMPUTE SERVICES 28](#_Toc124620613)

[AZURE STORAGE SERVICE 29](#_Toc124620614)

[AZURE STORAGE ACCOUNT 29](#_Toc124620615)

[DATA MIGRATION 67](#_Toc124620616)

[AZURE NETWORK SERVICES 68](#_Toc124620617)

[AZURE VIRTUAL NETWORK 68](#_Toc124620618)

[NETWORK SECURITY GROUP 71](#_Toc124620619)

[NETWORK CONNECTIVITY OPTIONS 73](#_Toc124620620)

[AZURE WEB-APP SERVICE 75](#_Toc124620621)

[AZURE APP SERVICE PLAN 75](#_Toc124620622)

[CREATING AN APP SERVICE 77](#_Toc124620623)

[DEPLOYMENT SLOTS 78](#_Toc124620624)

[AUTOSCALING THE AZURE WEB APP 81](#_Toc124620625)

[AZURE CONTAINER REGISTRY(ACR) 82](#_Toc124620626)

[CREATING AN ACR 82](#_Toc124620627)

[PUBLISH A DOCKER IMAGE IN ACR 83](#_Toc124620628)

[CREATING A CONTAINER INSTANCE 84](#_Toc124620629)

[AZURE KUBERNETES SERVICES 84](#_Toc124620630)

[KUBERNETES FEATURES 84](#_Toc124620631)

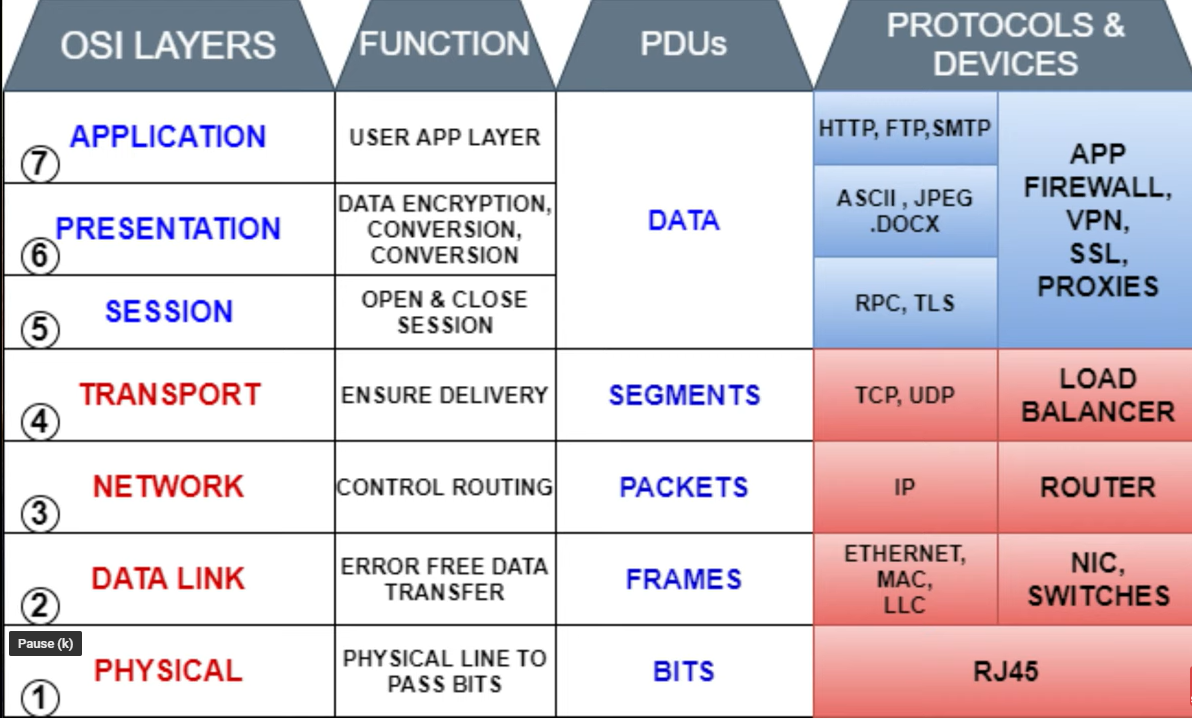
[KUBERNETES ARCHITECTURE 85](#_Toc124620632)

# CLOUD COMPUTING

## OSI MODEL

**The Open Systems Interconnection (OSI) model describes seven layers that computer systems use to communicate over a network.**

|  |  |
| --- | --- |
| Layers of OSI Model - GeeksforGeeks |  |



## SERVER TYPES

* Rack Servers
* Tower Servers
* Blade Servers

# BACKGROUND OF CLOUD COMPUTING

## ISSUES WITH IN-PREMISE INFRASTRUCTURE

When we have to set up a on premise infrastructure. There are multiple factors which we need to account for

1. **COST(EXPENDITURE) -**High CapEx and OpEx
2. **MAINTENANCE -** Hardware and Software maintenance
3. **SCALABILITY –** For scalability – the in-premise set up has to make sure that they have enough infrastructure available (processing power, Storage capacity etc..) to meet the high demand

### COST TYPES - 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.

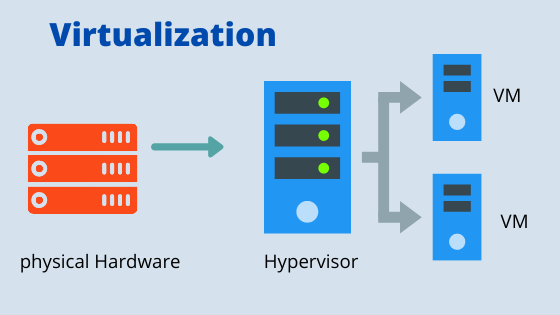
A consumption-based model has many benefits, including:

* 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.

## SOLUTION OF ISSUES OF IN-PREMISE INFRASTRUCTURE

To solve the issue, we have in premise infrastructure – industry moved towards a technology called **Virtualization**. The problem can be solved by setting up highly virtualized environment.

## 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.

### VIRTUALIZED ENVIRONMENT

|  |  |
| --- | --- |
|  | * In a highly virtualized environment – when have 2 or more physical servers connected to SAN (Storage Area Network). Collectively they are called Cluster * Each physical hardware is virtualized using a software called hypervisor (Type 1 or Type 2) * To manage the cluster – we need a Cluster Management Software which is installed on a dedicated server. * These cluster management software provides an interface (Portals) to manage the storage, VM, network etc.. * The examples of Cluster Management software are System Center, OpenStack or VMWare VSphere Suite.   **THIS ENTIRE SET IS CALLED PRIVATE CLOUD** |

### BENEFITS OF VIRTUALIZATION

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

### TYPES OF HYPERVISORS

#### TYPE-1 HYPERVISOR

* The hypervisor runs directly on the underlying host system. It is also known as a “**Native Hypervisor” or “Bare metal hypervisor**”. It does not require any base server operating system. It has direct access to hardware resources. Examples of Type 1 hypervisors **include VMware ESXi, Citrix XenServer, and Microsoft Hyper-V hypervisor**.
* Such kinds of hypervisors are very efficient because they have direct access to the physical hardware resources (like CPU, Memory, Network, and Physical storage). This causes the empowerment of the security because there is nothing any kind of the third-party resource so that attacker couldn’t compromise with anything.
* One problem with Type-1 hypervisors is that they usually need a dedicated separate machine to perform their operation and to instruct different VMs and control the host hardware resources.

#### TYPE-2 HYPERVISOR

* A Host operating system runs on the underlying host system. It is also known as ‘**Hosted Hypervisor**”. Such kind of hypervisors doesn’t run directly over the underlying hardware rather they run as an application in a Host system (physical machine). Basically, the software is installed on an operating system. Hypervisor asks the operating system to make hardware calls. **An example of a Type 2 hypervisor includes VMware Player or Parallels Desktop.**
* Such kind of hypervisors allows quick and easy access to a guest Operating System alongside the host machine running. These hypervisors usually come with additional useful features for guest machines. Such tools enhance the coordination between the host machine and the guest machine.
* Here there is no direct access to the physical hardware resources so the efficiency of these hypervisors lags in performance as compared to the type-1 hypervisors, and potential security risks are also there an attacker can compromise the security weakness if there is access to the host operating system so he can also access the guest operating system.

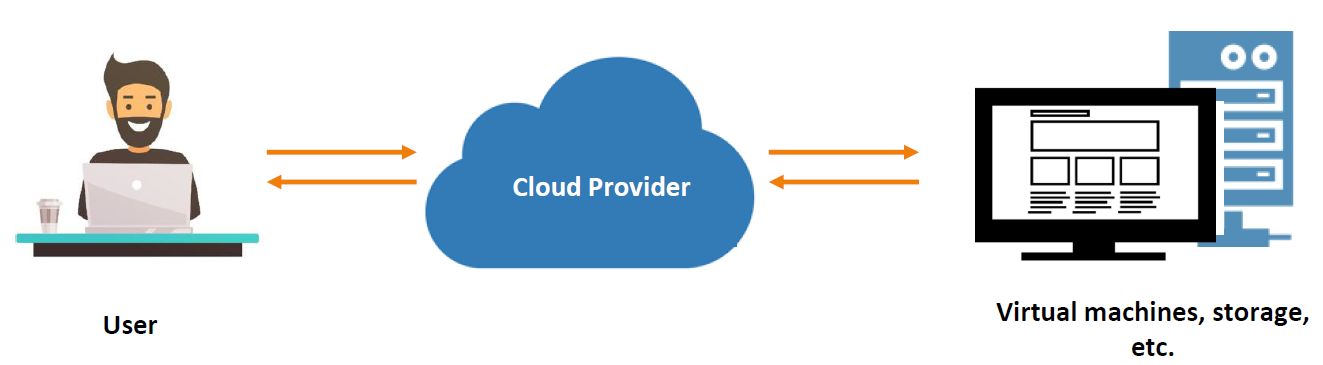
**Type 1 hypervisors offer much better performance than Type 2 ones because there’s no middle layer, making them the logical choice for mission-critical applications and workloads.**

# CLOUD CONCEPTS

To understand what cloud computing concept we is needed to understand the problem - it is solving for a traditional applications

|  |  |
| --- | --- |
| **SCALABILILTY** | * For a tradition application we need to increase hardware as user demand increases. * Need to increase staff to maintain the server   **Cloud computing: It helps in scaling up / down to meet the demand hence can able to solve the scalability issues.** |
| **COST** | High upfront expenses and capital expenditure; high maintenance cost  **Cloud computing - User has to pay only for the resources they use** |
| **MAINTENANCE** | Servers need to be managed and monitored continuously  **Cloud Computing: Servers are maintained by the cloud providers** |
| **SLOW DATA RECOVERY** | Takes more time to recover data; requires high budget and resources  **Cloud Computing : Fast data recovery and automated data backup** |

## 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:

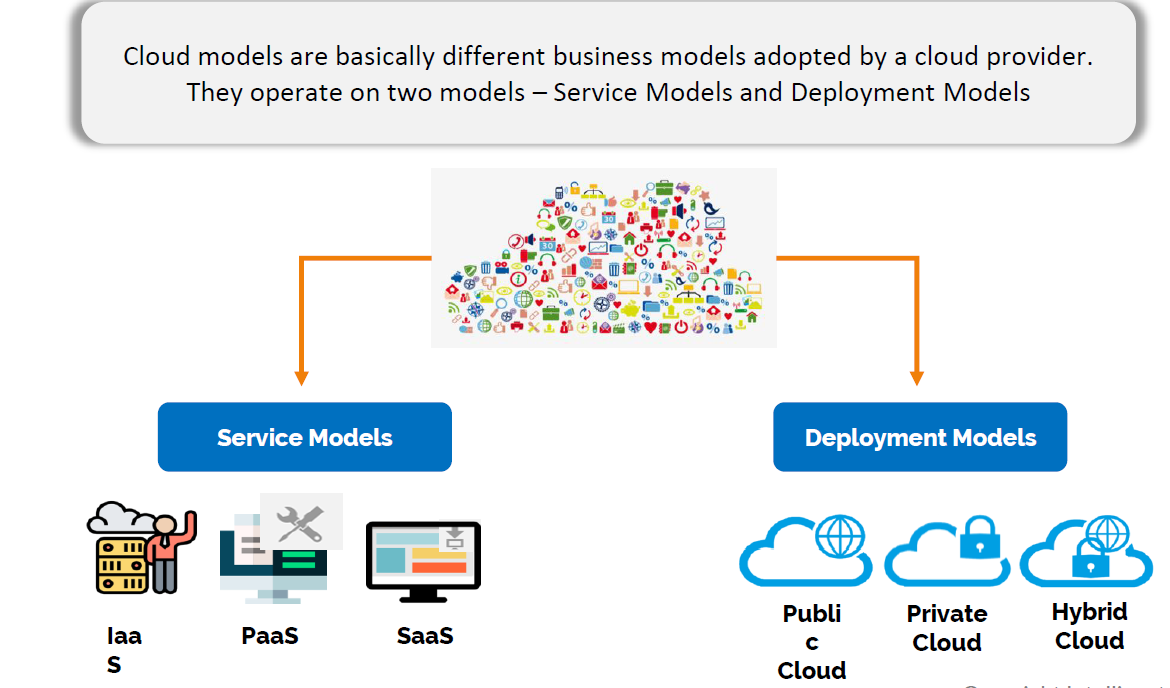
* Lower your operating costs.
* Run your infrastructure more efficiently.
* Scale as your business needs change.

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.

## BENEFITTS OF CLOUD

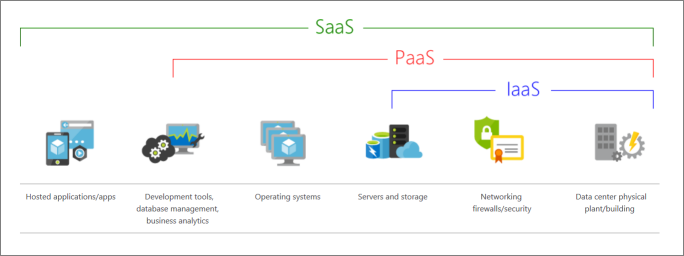
* **HIGH AVAILABILITY**: Depending on the service-level agreement (SLA) that you choose, your cloud-based apps can provide a continuous user experience with no apparent downtime, even when things go wrong.
* **SCALABILITY**: Apps in the cloud can scale vertically and horizontally:
  + Scale vertically 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.

## TYPES OF CLOUD MODELS



### 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 or Google Drive |

#### 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.

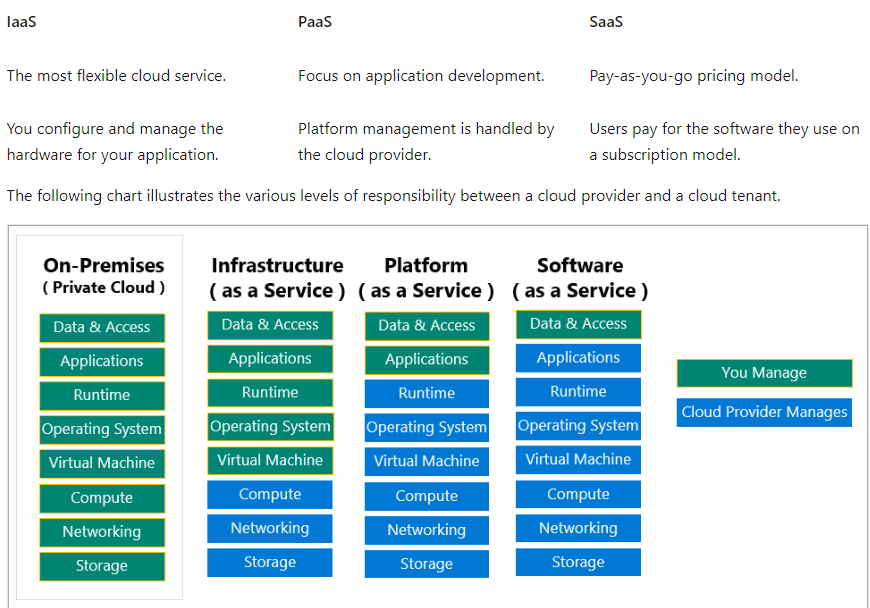
##### ADVANTAGES

* **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



### DEPLOYMENT 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. |

### WHAT IS SERVERLESS COMPUTING?

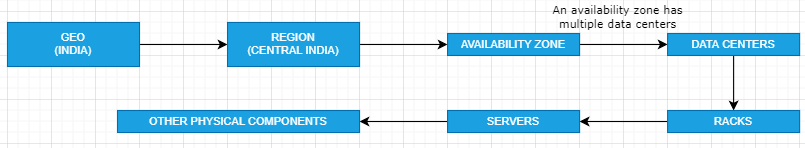
* Like PaaS, serverless computing enables developers to build applications faster by eliminating the need for them to manage infrastructure.
* With serverless applications, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code.
* Serverless 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?

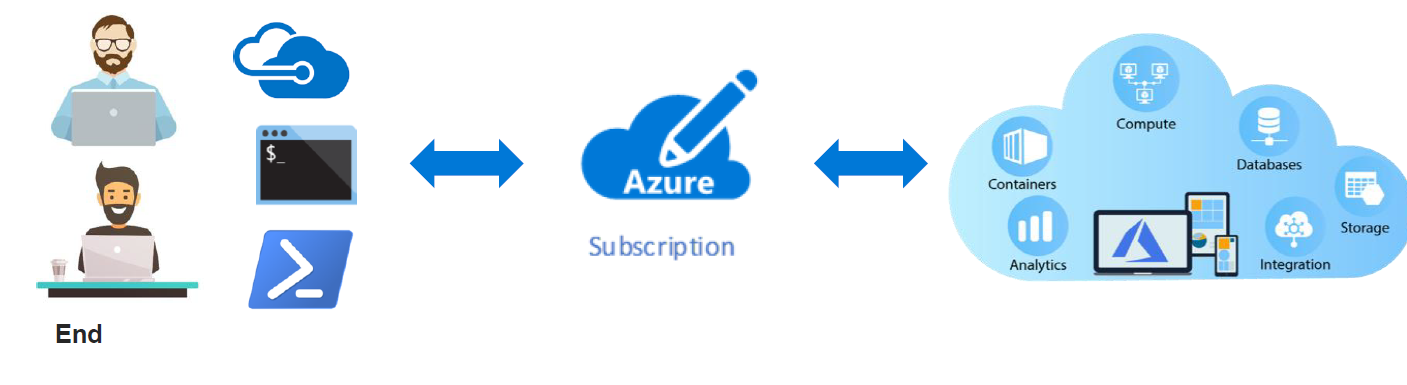
* Azure uses a technology known as **virtualization**. ***Virtualization separates the tight coupling between a computer's hardware and its operating system, using an abstraction layer called a hypervisor***.
* The hypervisor emulates all the functions of a real computer and its CPU in a virtual machine, 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 INFRASTRUCTURE SET UP



## AZURE SUBSCRIPTIONS, MANAGEMENT GROUPS, AND RESOURCES

* Azure subscription is an active agreement between Microsoft and its users. This agreement provides users the needed access to avail the services and resources offered by Microsoft Azure.
* User can have multiple Azure subscription linked to its Azure Account.



|  |  |
| --- | --- |
| 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.*** |  |

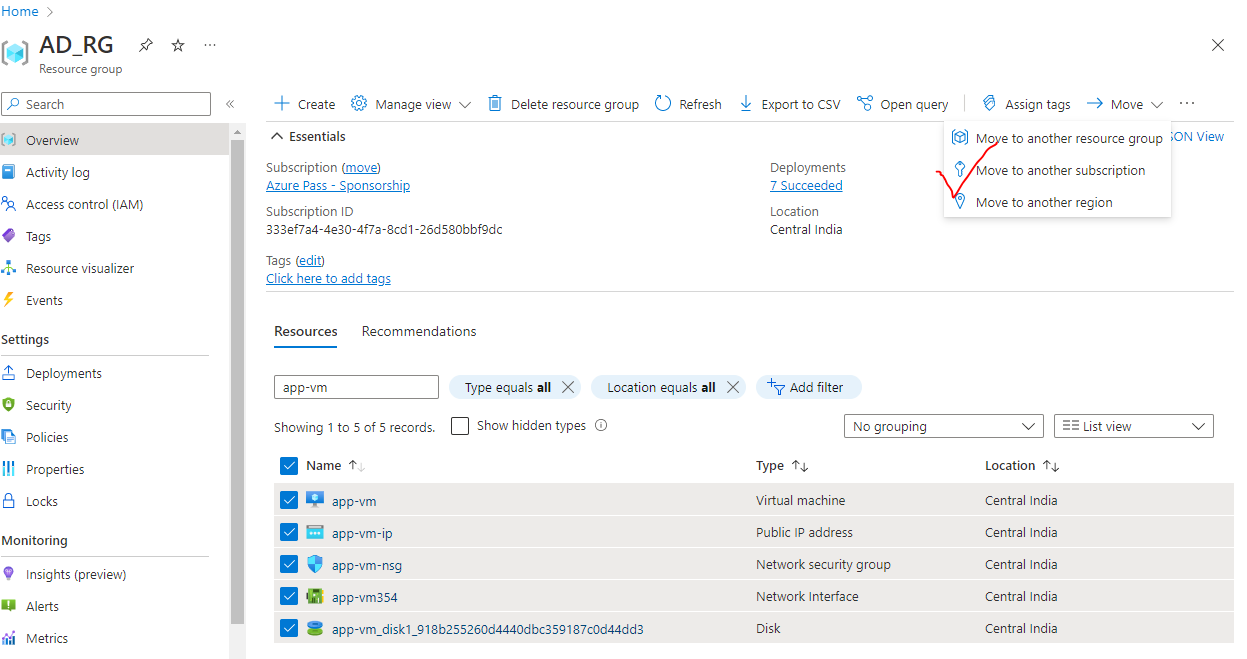
#### CREATING A RESOURCE GROUP USING AZURE CLI AND POWERSHELL

### MOVING THE RESOURCES ACROSS RESOURCE GROUP

|  |  |
| --- | --- |
|  | * Resources can be moved from one resource group to another. Note – Moving the resources does not change the location of the resource – it has nothing to do the region of resource group. |

### MOVING THE RESOURCES ACROSS SUBSCRIPTION

* We can move the resource across the subscription.
* When we want to move the resource to another subscription, then we must move its dependent resources as well onto that subscription.
* For example for a VM, this includes disk, Azure virtual network, the public IP address, etc.
* Go the resource group 🡪 Select all the dependent resources 🡪 Move to another subscription



### RESOURCE TAGS

|  |  |
| --- | --- |
|  | * Resource tags are used to organize the resources * Each tag consists of a name on a value pair. For example, if we want to organize the resources which belong to a specific department in an organization – we can make use of tags * To create a resource tag 🡪 Go to the resource * Using Tags – we can be able to filter the resources tags * We can get the cost analysis of the resources based on tags. For example – Cost analysis of tags -which belongs to a department |
|  |

### RESOURCE LOCKING

* Looking resources helps ensure users don’t accidently delete or modify resources

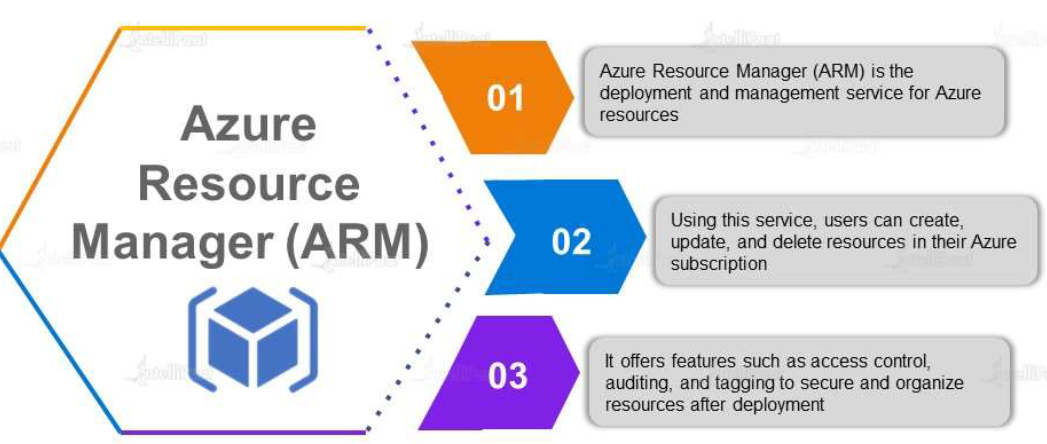
#### TYPES OF RESOURCE LOCKS

|  |  |
| --- | --- |
| ***CanNotDelete*** | Authorized users can read and modify the resources, but they can’t delete the resources |
| ***ReadOnly*** | * Authorized users can read the resources, but they can’t delete or update the resources * For example – for the VM having ready only lock- We cannot be able start or stop the VM – because read only lock does not even allow to update the properties of the resource.(status) |

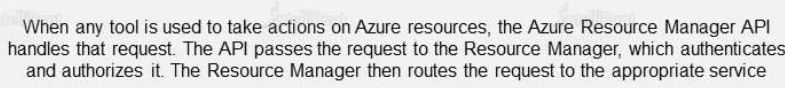
#### LAB – LOCKING RESOURCES

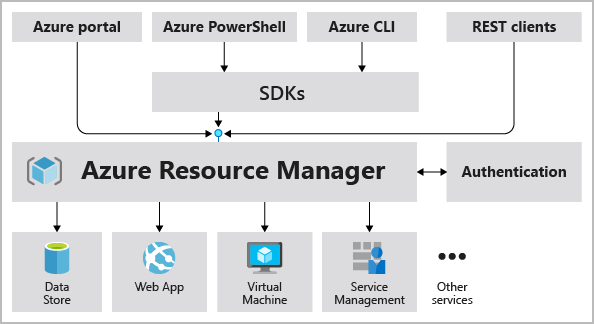
|  |  |
| --- | --- |
|  | ***Lock can be applied on resouce level , resource group level and subscription level***   * Any lock on resource level will lock the given resources. * But a lock on resource group level it will inherited by all the resources in that resource group. * But a lock on subscriptionlevel it will inherited by all the resources in that subscription. * To remove the lock on any resource – we have to delete the lock itself |
| LOCk applied on Resource Group level | Inherited Lock (by the resources) |

### AZURE RESOURCE MANAGER (ARM)



#### HOW ARM WORKS?



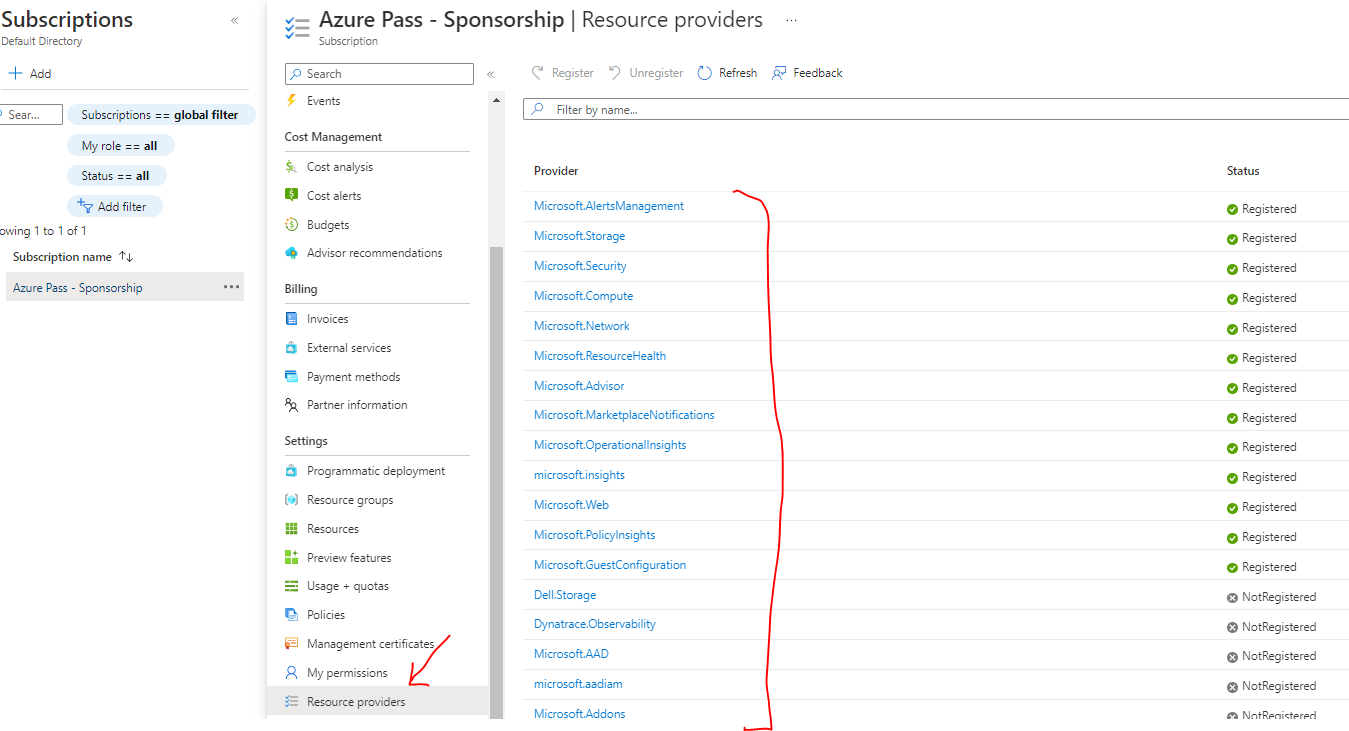


* When a user sends a request from any of the Azure tools, APIs, or SDKs, (***Note – From PowerShell & CLI the request to ARM goes via SDK***)
* When a Resource Manager receives the request for a resource. It first authenticates and authorizes the request.
* And if we have required permission - ARM sends the request to the Resource Providers

**NOTE –**

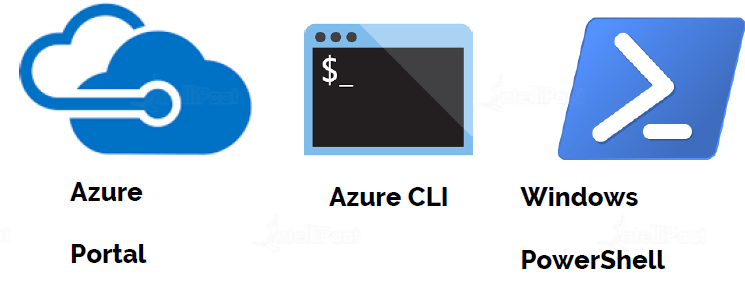
* + The resource provider details can be found in Subscription 🡪 Resource Provider
  + Example: if we are requesting to create a Storage Account. The request will go to **ARM 🡪 Microsoft.Storage(Provider). Hence - The Resource Provider is one which do the heavy lifting**

##### AZURE RESOURCE PROVIDER



## ACCESSING AZURE PLATFORM

Microsoft provides various ways to access the Microsoft Azure platform. For those who prefer a GUI, there is Azure Classic Portal. For those who prefer command-line tools, there is Azure PowerShell or Azure CLI



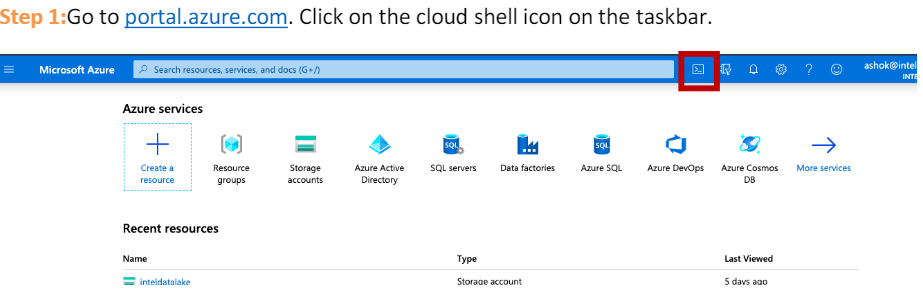
|  |  |
| --- | --- |
| **PORTAL** | Azure Portal is a web-based, unified console that lets users access and manage Azure services. Using Azure Portal, users can build, monitor, and manage their applications on Azure Cloud. To sign into the portal, users need to have an Azure account |
| **WINDOWS**  **POWERSHELL** | Azure PowerShell is a task-based command-line shell built on the .NET framework. It lets users control Azure's robust functionality from a command li |
| **AZURE CLI** | Azure CLI is a cross-platform command-line tool used to manage and monitor Microsoft Azure platform and services. It provides an alternative for PowerShell |

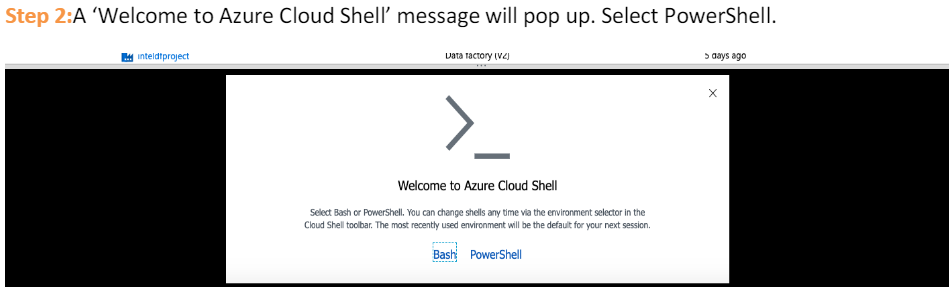
* Azure CLI is meant to be used with Azure only. But powershell can be used with Microsoft 365 or windows based system as well

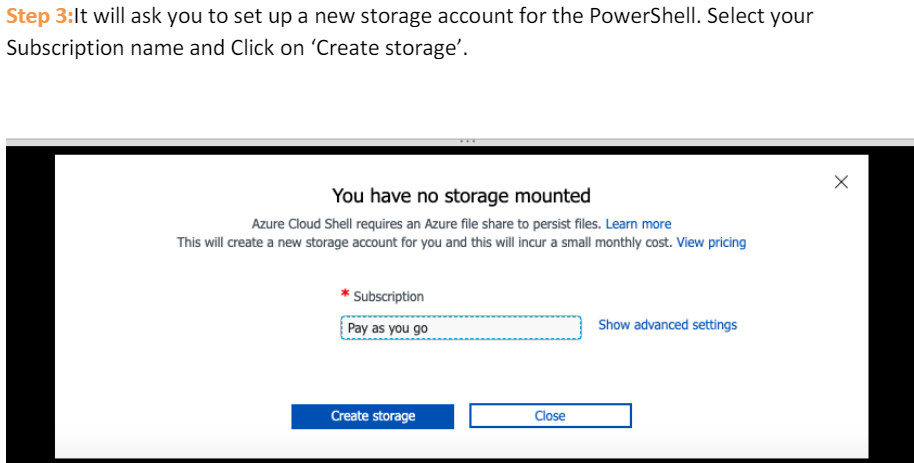
### POWERSHELL

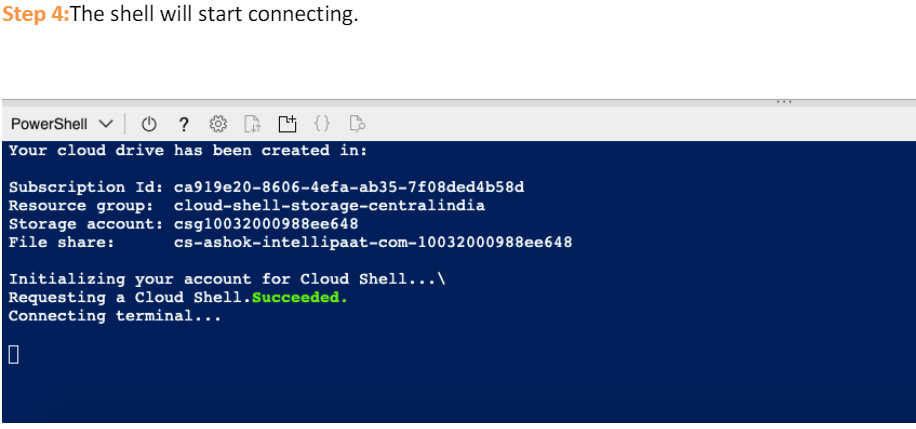
* Powershell gives a feature of
  + COMMAND LINE SHELL
  + SCRIPTING LANGUAGE
  + CONFIGURATION FRAMEWORK
* It can run on *Linux , windows and MacOS*
* Usually in a command line tool for text input it gives a text output . unlike other command line tool- Powershell has ability to work with .Net objects
* The powershell commands are called Cmdlets. These Cmdlets are organized into Modules.
* In the basic installtion of powershell we don’t get all the modules- hence we have to install required modules manually

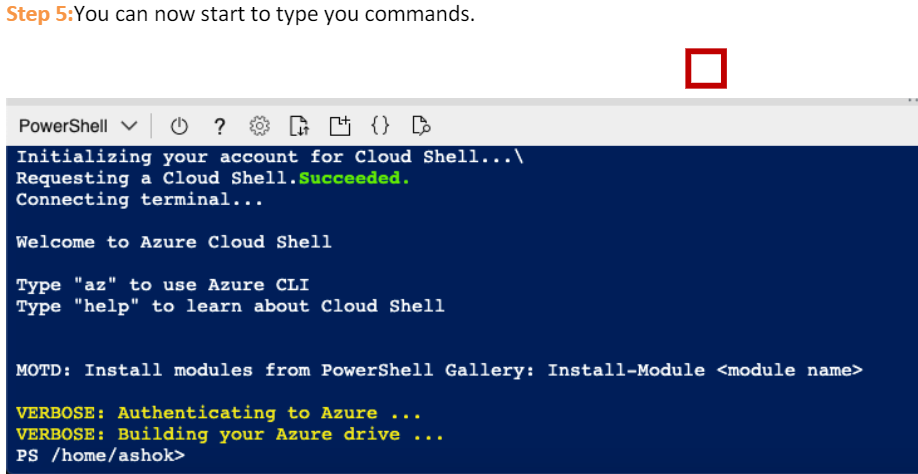
|  |
| --- |
| TO INSTALL THE MODULE FOR AZURE CMDLETS- FOLLOW THE FOLLWING STEPS |
| ###########BELOW STEPS ALLOWS YOU TO SETUP AZURE POWERSHELL ON YOUR WINDOWS LAPTOP################  ######################STEPS FOR MAC AT THE END OF THE FILE########################################  ###STEP 1: Always start Powershell ISE > Run AS Administrator###  ###STEP 2: Validate PowerShell Version is above 5###############  $PSVersionTable  ###STEP 3: if PowerShell version is above 5, Proceed to next steps. If not, install latest powershell by going to the link below###  ###https://www.microsoft.com/en-us/download/details.aspx?id=54616###  ###STEP 4: Enable script execution policy(This is disabled by default in Windows 10 and Above)###  Set-ExecutionPolicy -ExecutionPolicy RemoteSigned  ###STEP 5: Install Azure PowerShell Modules###  Install-Module -Name Az -AllowClobber #If any warning pops up, select yes, if asking for NuGET, select yes  ###STEP 6: Validate the modules have been installed###  Get-InstalledModule -Name Az -AllVersions | select Name,Version  ###STEP 7: Login to Azure from PowerShell###  Connect-AzAccount #If you see any errors like, unable to acquire token, use the below command instead  Connect-AzAccount -Tenant 'xxxx-xxxx-xxxx-xxxx' #Tenant ID of your account will be shown in the error, copy paste the same  ###STEP 8: Create a new resource group###  New-AzResourceGroup -Name TestRG -Location centralus  ###STEP 9: Remove Resource Group###  Remove-AzResourceGroup -Name TestRG -Force  ##################################END#####################################  #############################STEPS FOR MAC#############################################  #Follow Below link to install PowerShell, Download the package and install > You might need to allow the package installation from  #System Preferences > Security > Allow Install  #x64 processors - powershell-7.2.5-osx-x64.pkg https://github.com/PowerShell/PowerShell/releases/download/v7.2.5/powershell-7.2.5-osx-x64.pkg  #M1 processors - powershell-7.2.5-osx-arm64.pkg https://github.com/PowerShell/PowerShell/releases/download/v7.2.5/powershell-7.2.5-osx-arm64.pkg  #Once PowerShell has been installed, follow steps 5 - 9 from above |

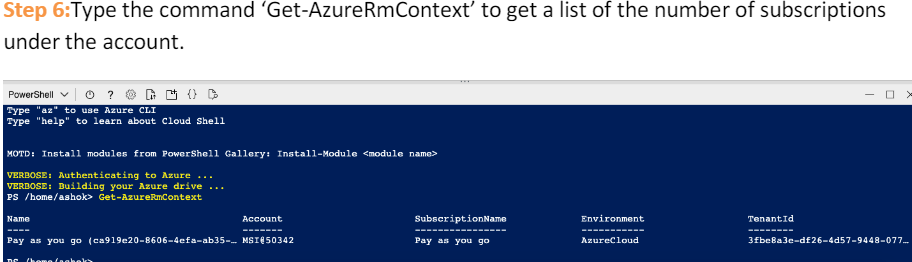












**POWERSHELL COMMAND LIST** - <https://learn.microsoft.com/en-us/powershell/module/az.resources/?view=azps-8.3.0>

#### POWERSHELL COMMAND

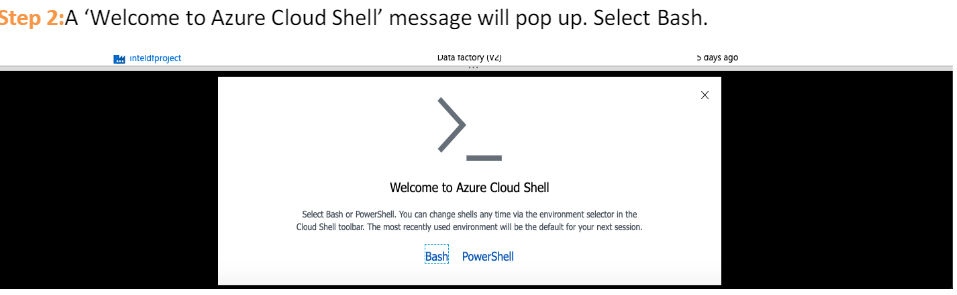
|  |  |
| --- | --- |
| **DETAILS** | **COMMAND** |
| ***New-AzResourceGroup -Name <name\_of\_res\_grp> -Location <loc\_of\_res\_grp>*** | * Creating a resource group in a given region |
| Get-AzResourceGroup | | * List all Resource Group |
| USING PIPE(FILTERING)  Get-AzResourceGroup | Where-Object {$\_.Location -eq "eastus"} | * The powershell command make use of pipeline(pipes) * Example – This command will find the Resource Group located in “East US” Region |
| Get-AzResourceGroup | Where-Object {$\_.Location -eq "eastus"}|Remove-AzResourceGroup -Force | * Delete All the resource group in a given location |

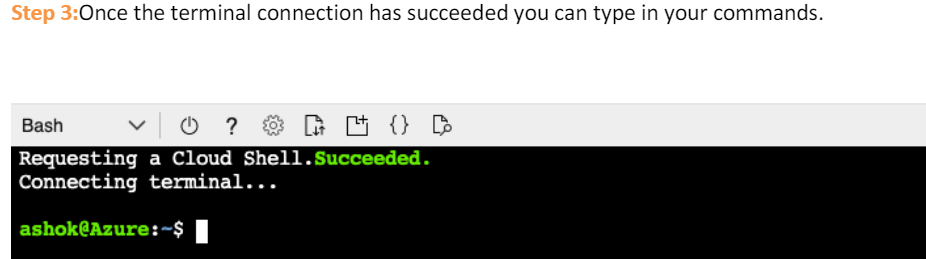
##### VARIABLES IN POWERSHELL SCRIPTS

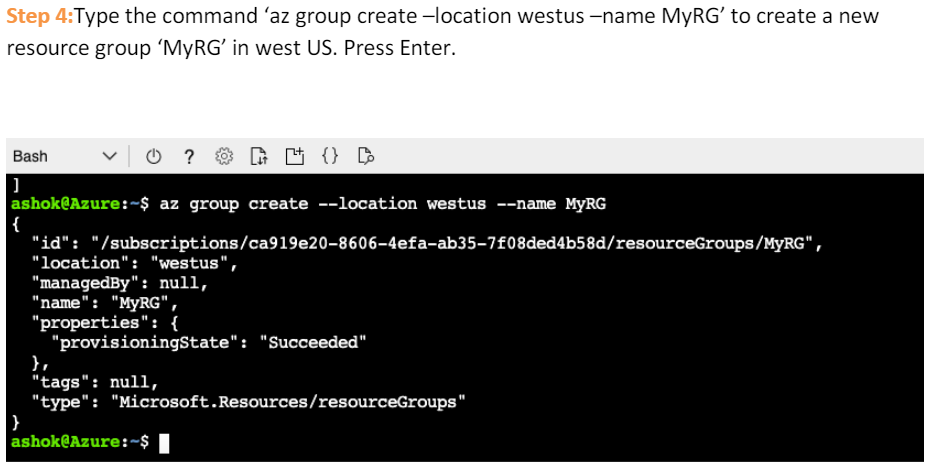
* We can create a script file to execute the Powershell commands

|  |
| --- |
| TO CREATE A RESOURCE GROUP IN A GIVEN REGION  **$resourceGrp="exam-grp"**  **$location="eastus"**  New-AzResourceGroup -Name $resourceGrp -Location $location |
| RUNNING A POWERSHELL SCRIPT IN VS CODE |

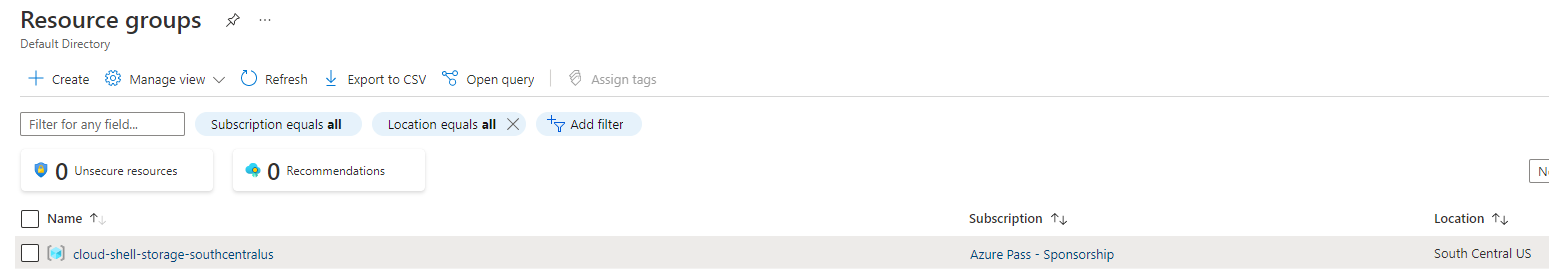
### AZURE CLI







RESOURCE GROUP CREATED



AZURE CLI COMMAND LIST - <https://learn.microsoft.com/en-us/cli/azure/group?view=azure-cli-latest>

#### AZURE CLI COMMAND

|  |  |
| --- | --- |
| **DETAILS** | **COMMAND** |
| ***az group create -l southcentralus -n new-rg*** | Creating a resource group in southcentralregion |
| az group list | List all Resource Group |
| **FILTERING**  az group list ***--query "[?location=='westus']"*** | List all resource groups located in the West US region. |

### 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 REGIONS 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

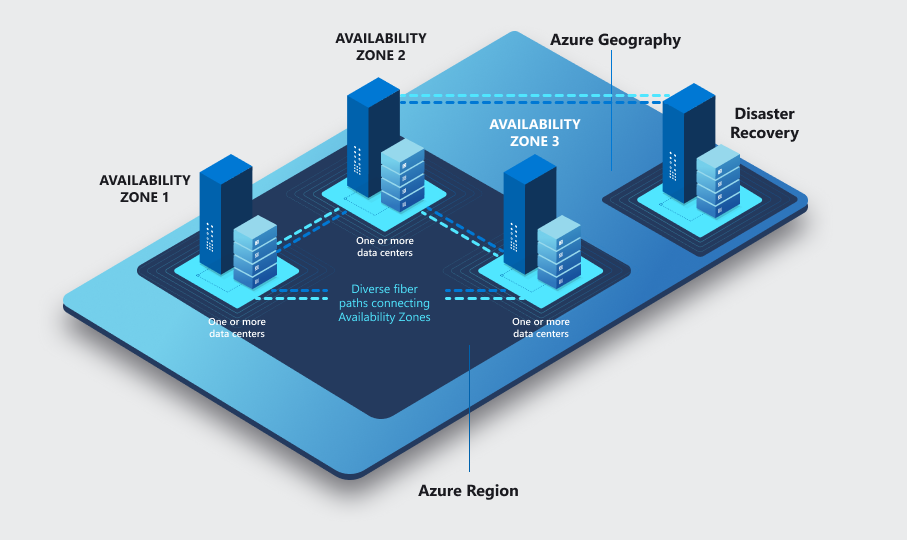
Azure has more global regions than any other cloud provider. These regions give us the flexibility to bring applications closer to users no matter where they are. Global regions provide better scalability and redundancy. They also preserve data residency for the services.

|  |  |
| --- | --- |
|  | CASE 1: SINGLE DATA CENTER IN A REGION  Imagine if application is deployed in a single 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? - the *application will go 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

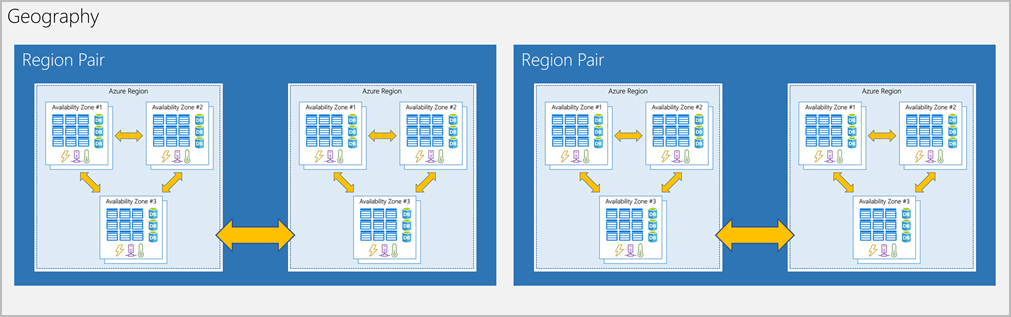


* Availability zones are physically separate datacenters within an Azure region.
* Each availability zone is made up of 3 more datacenters. Each datacenter is equipped with independent power, cooling, and networking.
* An availability zone is set up to be an isolation boundary. If one zone goes down, 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.***

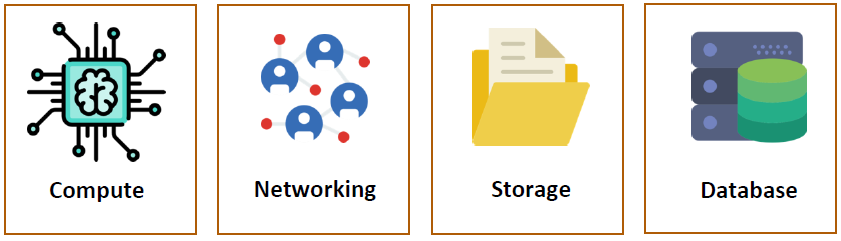


**ADDITIONAL ADVANTAGES OF REGION PAIRS:**

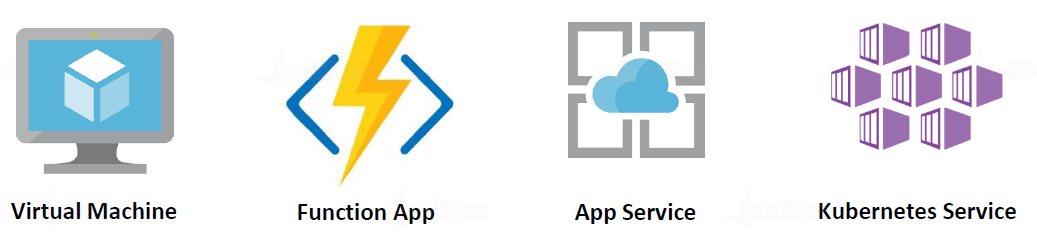
* If an extensive Azure outage occurs, one region out of every pair is prioritized to make sure at least one is restored as quickly as possible for applications hosted in that region pair.
* Planned Azure updates are rolled out to paired regions one region at a time to minimize downtime and risk of application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

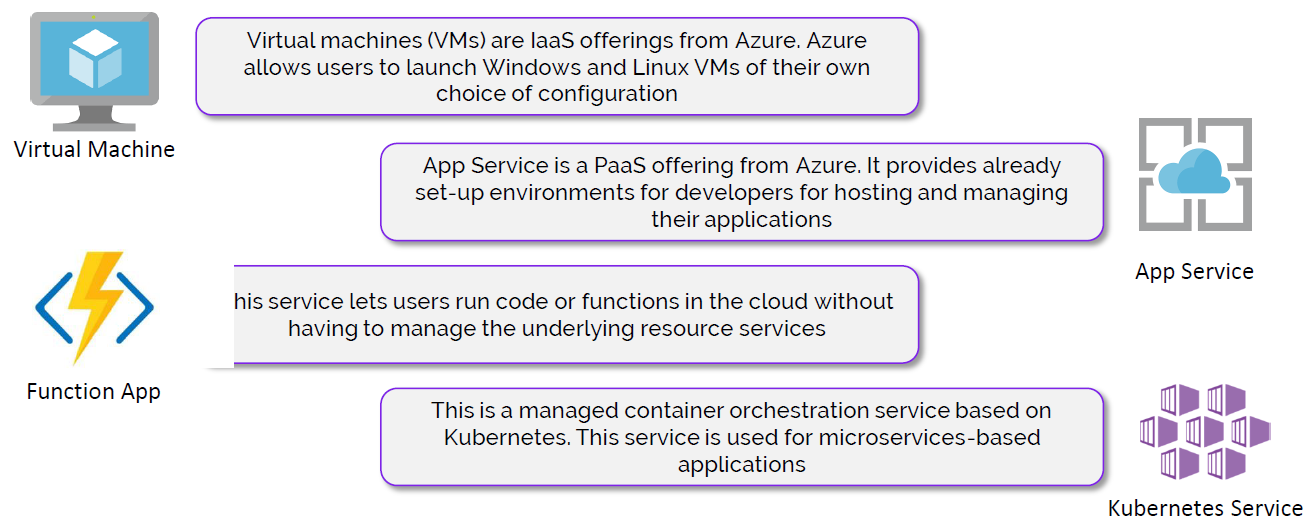
## AZURE SERVICES

Microsoft Azure offers various web-scale cloud services that are grouped together based on different business usage. These groups are called domains. Some of the main domains are listed below



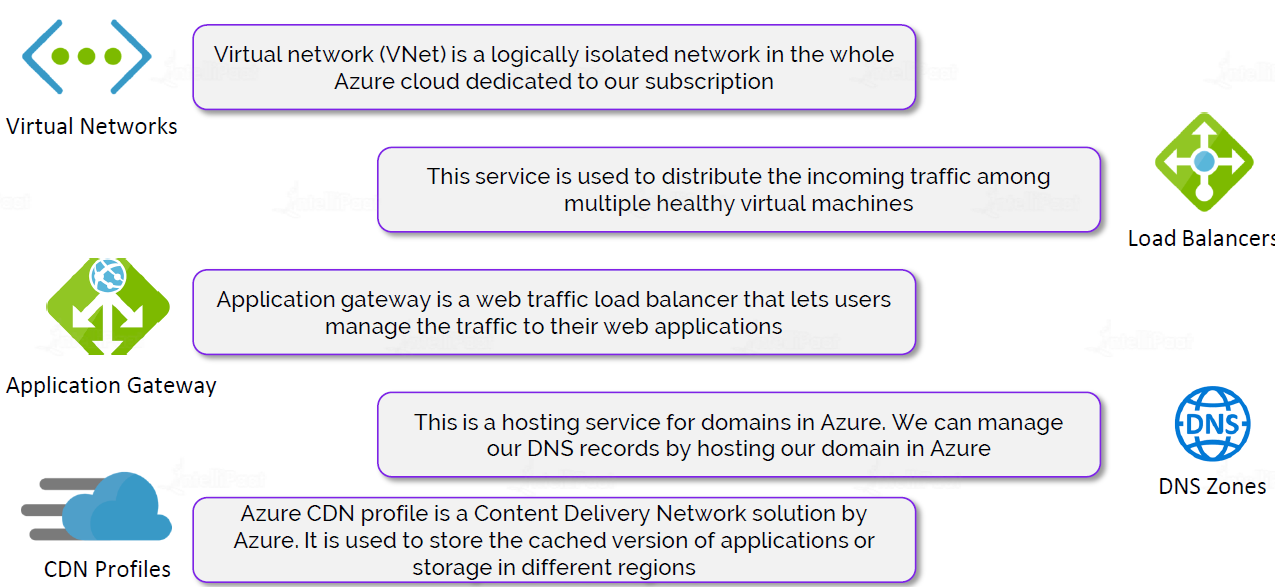
### COMPUTE SERVICES





### NETWORKING



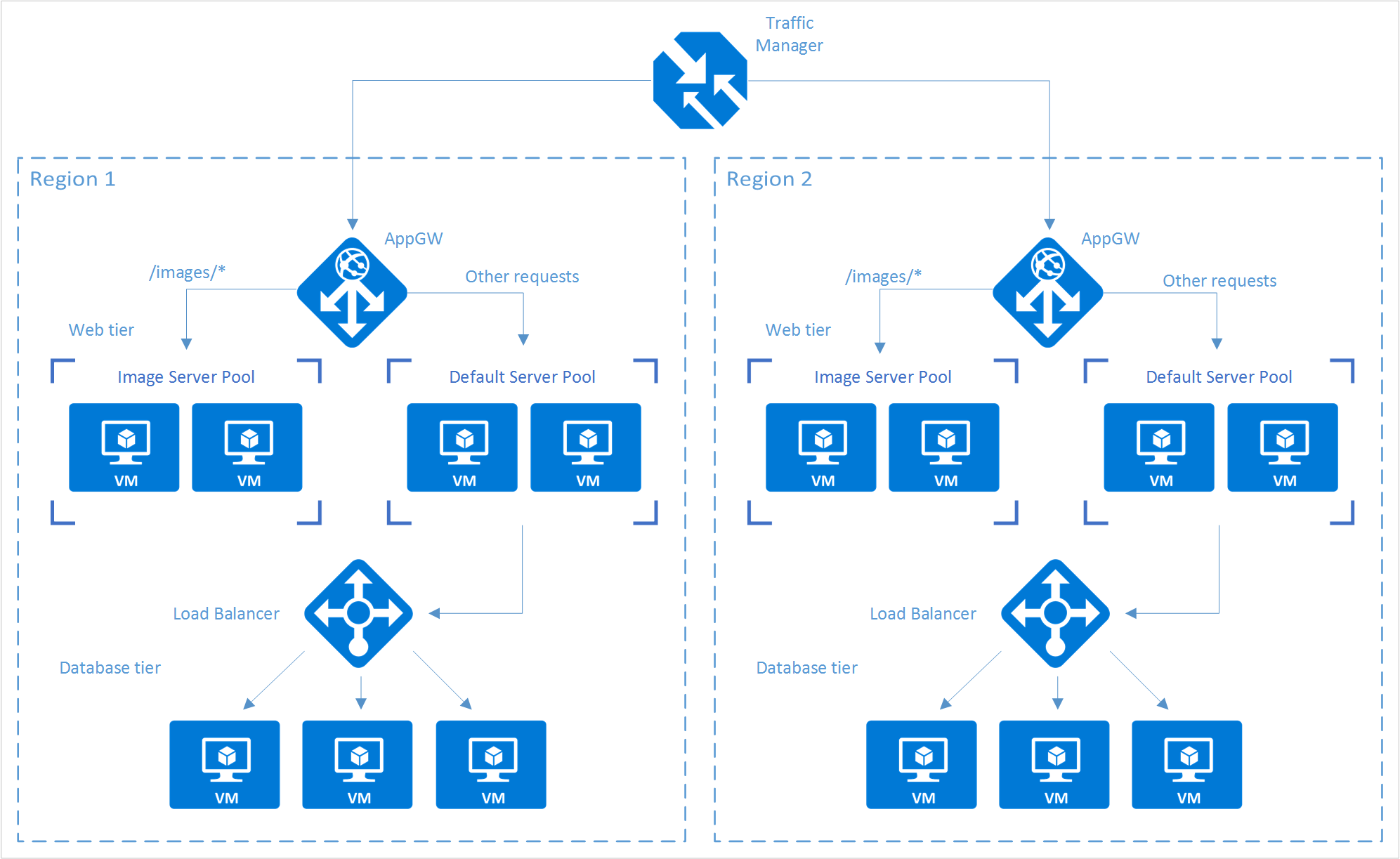


#### WHAT IS THE DIFFERENCE BETWEEN LOAD BALANCER AND APPLICATION GATEWAY?

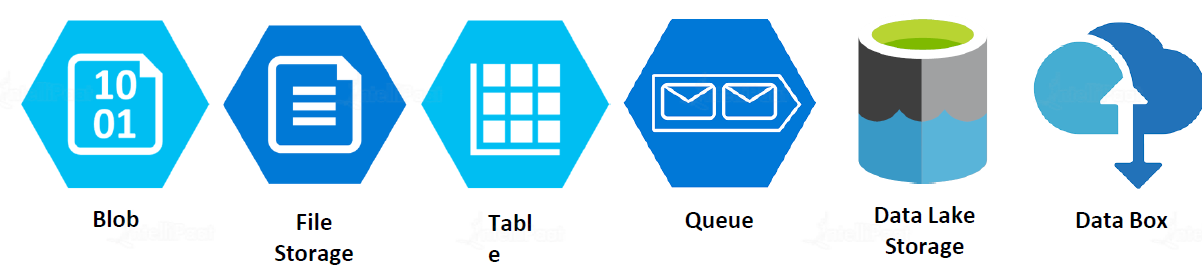
|  |  |
| --- | --- |
|  | * The Application Gateway work at application layer (Layer 7) but load balancer works at transport layer(Layer 4) * Application Gateway is used om top of load balancer to make routing based decision. Means we can make decision of routing the traffic based on type of request |

##### EXAMPLE

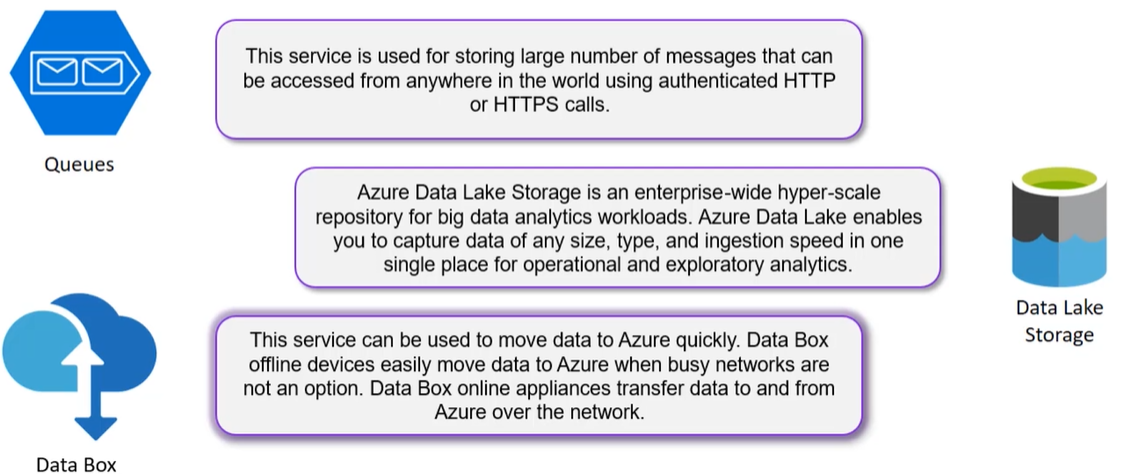
* Lets say – we have applicationwhere we have VM pools dedicated for “image” upload and another VM pool which handles “form” submission.
* Here we can distribute the traffic among the VMs using load balancer – but which type of request (image upload or form submission) will go which VM is decided by Application Gateway.



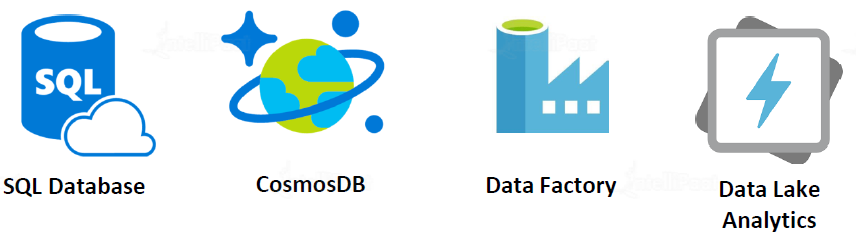
### STORAGE SERVICE

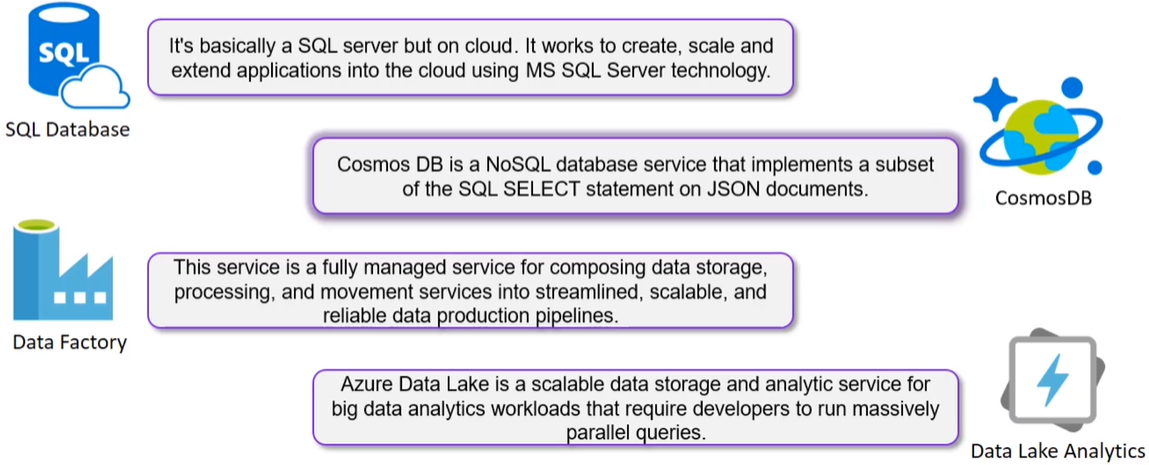






### DATABASE



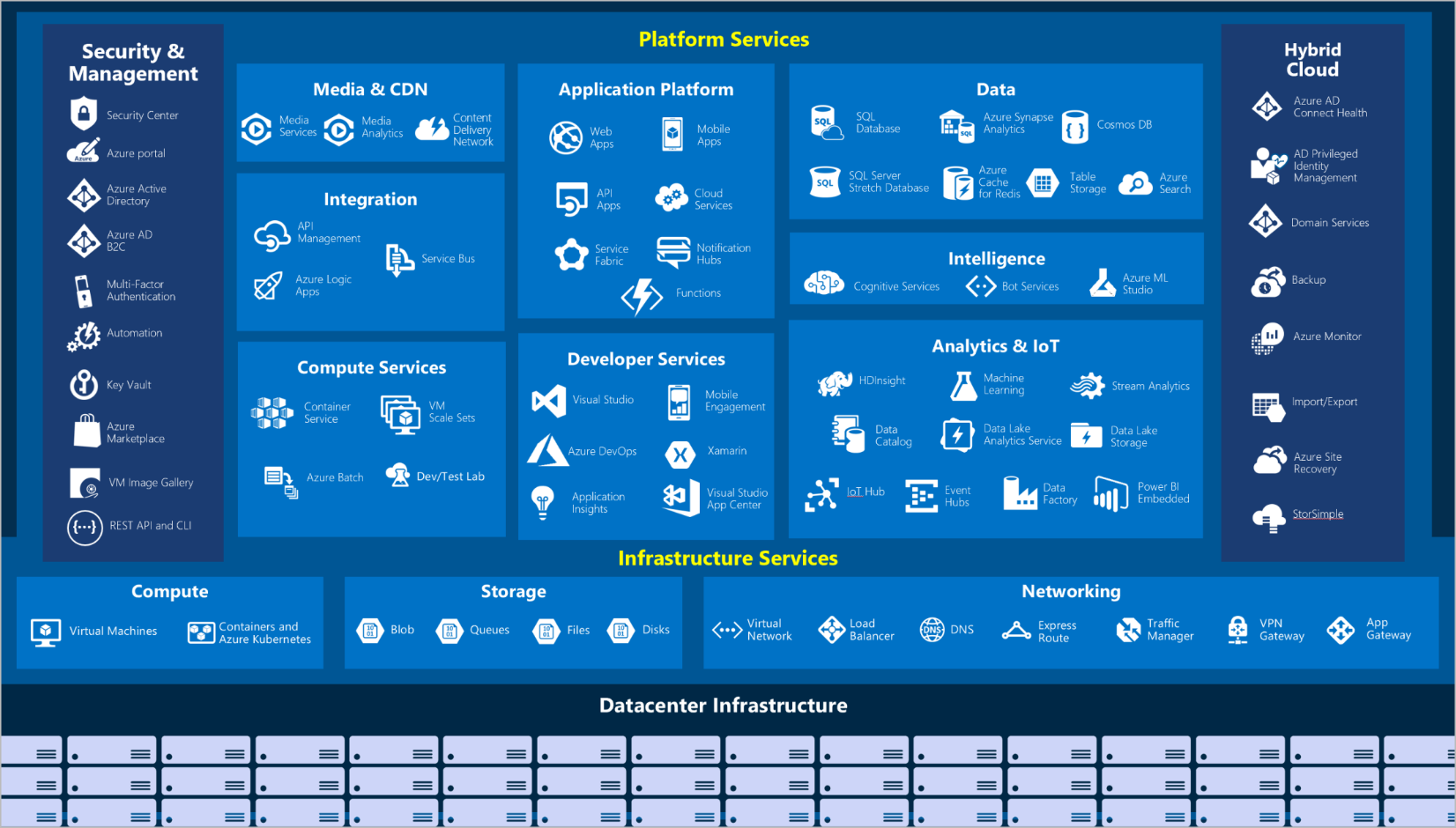


# AZURE CORE SERVICES

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**
  + **OPERATING SYSTEMS.**
* The resources are available on-demand. We pay only for the resources we use, and only for as long as we are using them.

|  |  |
| --- | --- |
| AZURE VIRTUAL MACHINES | * With Azure Virtual Machines, we can create and use VMs in the cloud. * ***Virtual Machines provides infrastructure as a service (IaaS)*** * We use VMs when we need total control over an operating system and environment. * Just like a physical computer, we can customize all the software running on the VM. This ability is helpful when we are 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 can be used to deploy and manage a set of identical VMs. * With all VMs configured the same, virtual machine scale sets are designed to support autoscaling. 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, we can quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. * We 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) | * Azure Functions are ideal when we are concerned only about the code running your service and not the underlying platform or infrastructure. * They're commonly used when we need to perform work in response to an event (often via a REST request), timer, or message from another Azure service. |
| 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 we 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. * We can run multiple instances of a containerized application on a single host machine. |

## AZURE NETWORK SERVICES

### AZURE VIRTUAL NETWORK

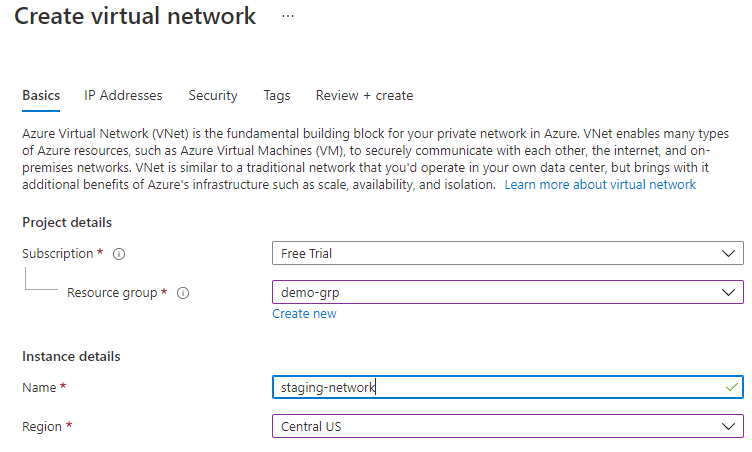
* The Virtual network is a is always a range of IP address.
* When we create a VM it is always get a private IP address. If the private IP of the VM fall in the range of virtual machine, then the machine will be part that virtual network. Hence, The private IP helps in locating the VM in the vitual network.
* 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

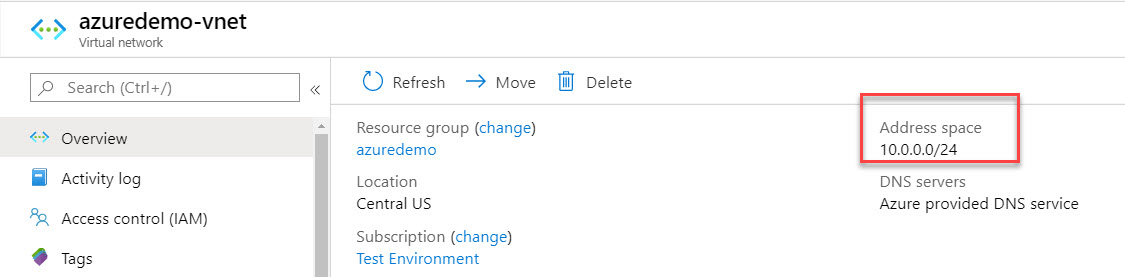
* Usually the Virtual network is created aling with the VM. But we can also able to create Virtual machine upfront and the tie the VM into it.



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



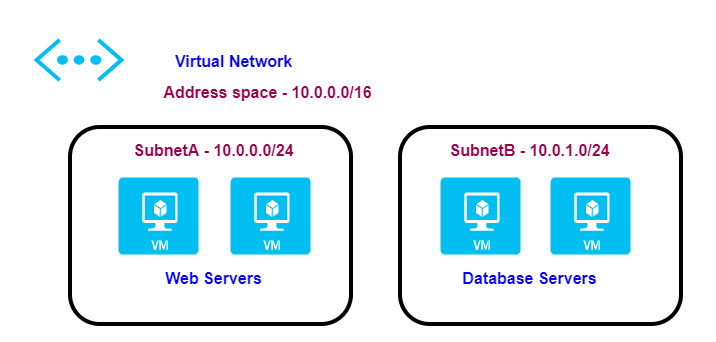
* *Note: When we create a VM needs to be part of virtual network*
* Virtual network has something called IP address range / address space as shown below

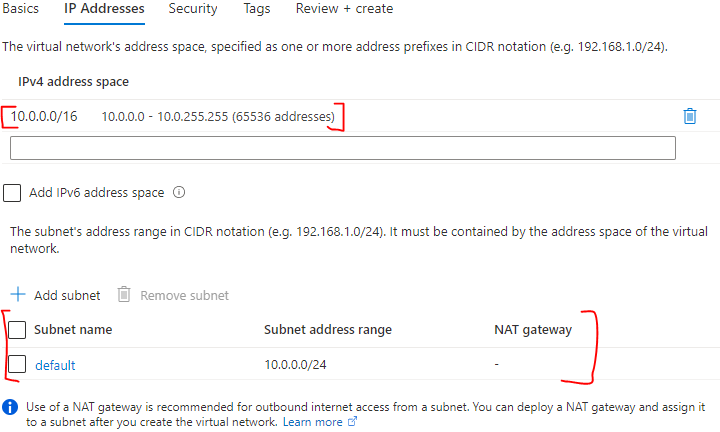


#### CREATING A SUBNET IN VIRTUAL NETWORK

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

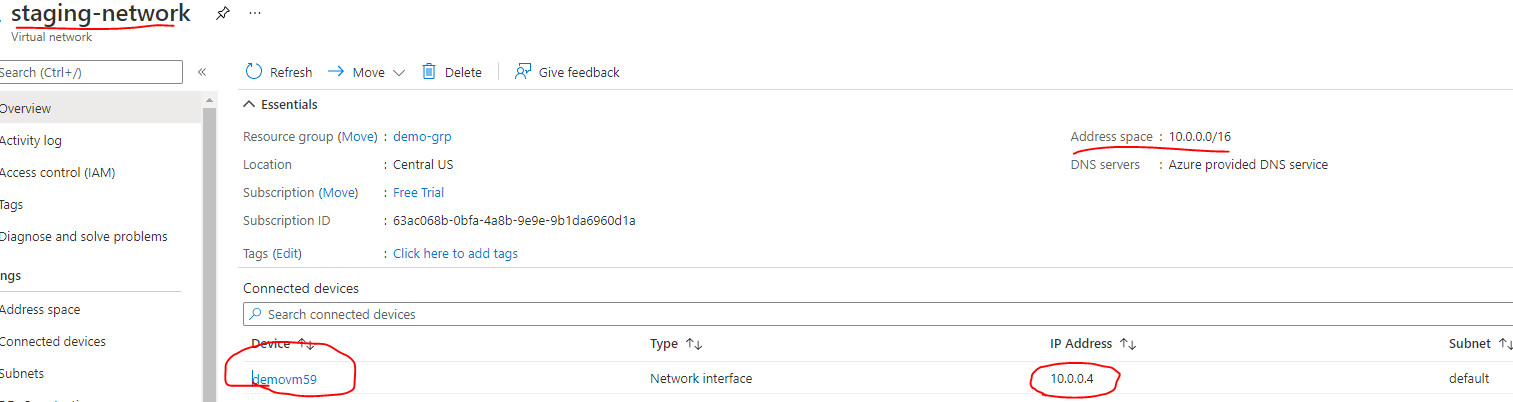
An example is shown below of having multiple subnets. You could have one subnet named SubnetA in the virtual network to host your Web servers and another subnet to host the Database servers.





#### 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.



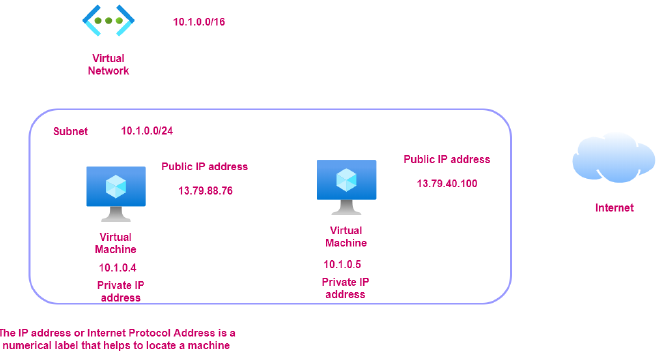
* The above diagram shows the VM created in the virtual network.
* The IP of the VM lies within the range of Virtual Network

*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.

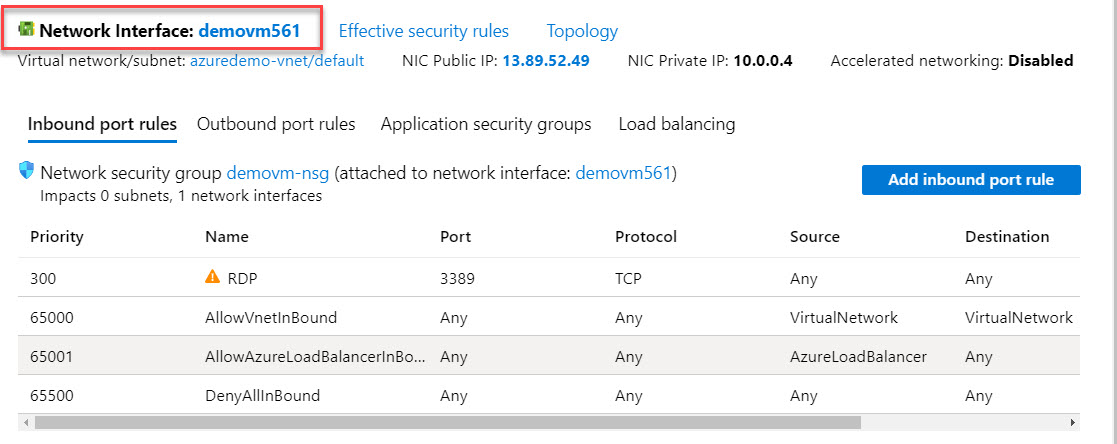


### NETWORK SECURITY GROUP

* 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



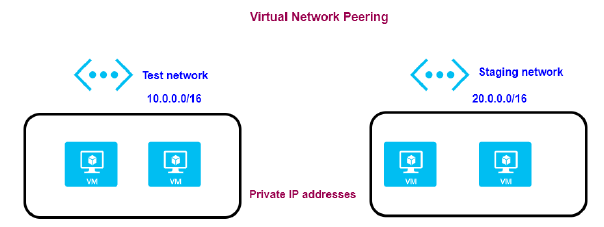
#### APPLICATION SECURITY GROUP

|  |  |
| --- | --- |
|  | * 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

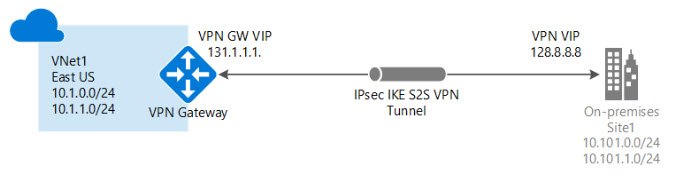
#### **VIRTUAL NETWORK PEERING**

* 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.



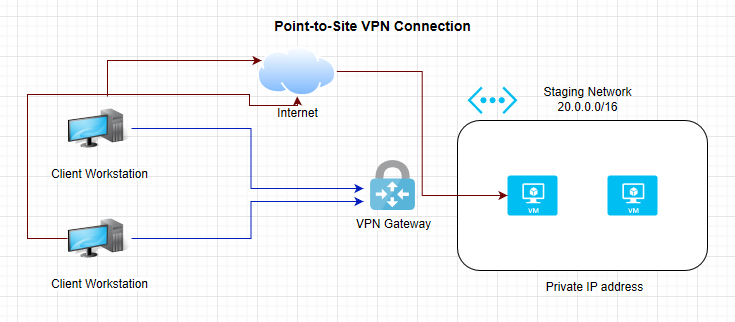
* On the on-premise side, you need to have a VPN device that can route traffic via the Internet onto the VPN gateway in Azure. The VPN device can be a hardware device like a Cisco router or a software device ( e.g Windows Server 2016 running Routing and Remote services). The VPN device needs to have 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**

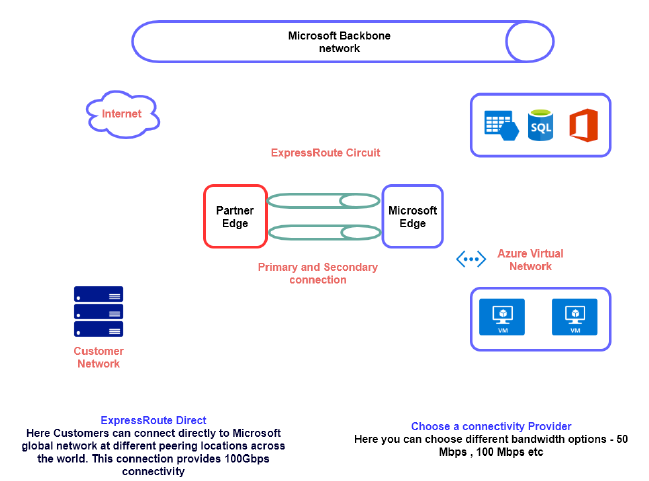
* A Point-to-Site VPN connection is used to establish a secure connection between multiple client machines and an Azure virtual network via the Internet.

##### 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



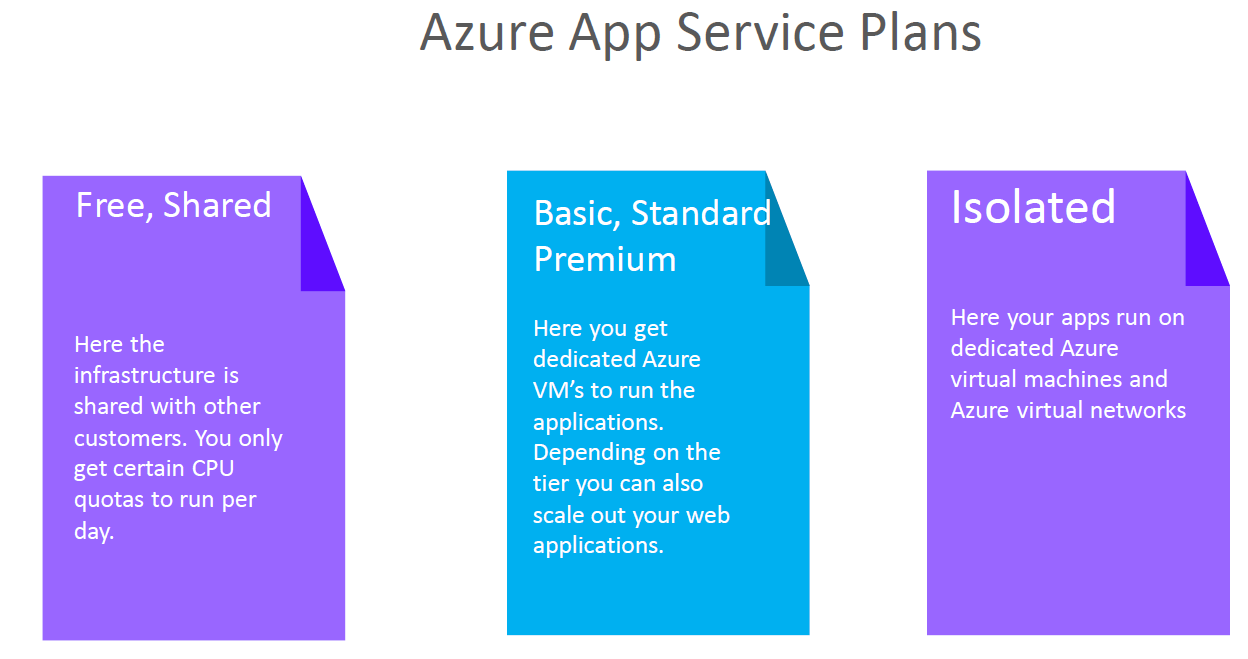
## AZURE WEB-APP SERVICE

|  |  |
| --- | --- |
| * Web App Service is a platform-as-a-service where the undelying infrastructure is managed ny Azure itself. * This is an HTTP-based service used for hosting web applications. * The web application can developed in languges like .NET, .NET Core, Java, Ruby, Node.js or Python which can run on Windows or Linux-based platforms. * Since is a PAAS, the application gets deployed on an underlying compute infrastructure (VM). The entire infrastructure is managed by Azure themselves i.e., by the Azure App Service. * Azure webapps has a features such as auto scaling and security and DevOps capabilities |  |

### AZURE APP SERVICE PLAN

* App service plan defines the set of compute resources that are used to run the web application
* When we create a web app in Azure, we need to create something known as the App Service Plan. Hence the Web app service are always linked to an app service plan.
* The billing of an app service depends upon the type of app service plan that we choose ( <https://azure.microsoft.com/en-us/pricing/details/app-service/windows/#pricing> )

#### TYPES OF APP SERVICE PLAN





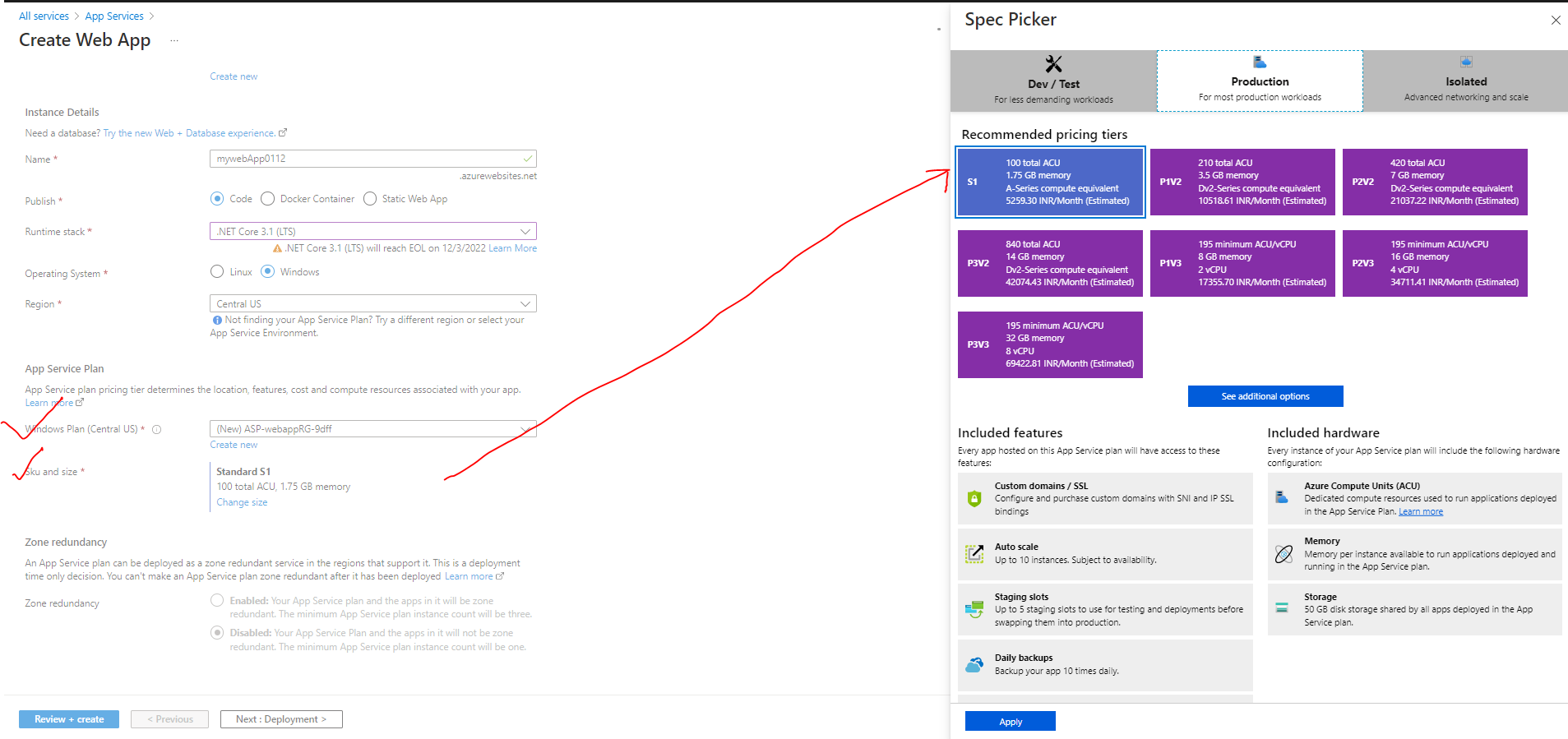
* In the above pricing table of app service plan show different features associated with different App service plans.

##### UNDERDTANDING THE BASIC APP SERVICE PLAN

|  |  |
| --- | --- |
|  | * In Basic plan we can deploy unlimited web app and storage capacity of 10 GB in total. We can be able to scale up to 3 VM instances. * Let's say we create an Azure app service plan (basic pricing tier). * In this plan - we can create unlimited Azure Web applications and link it to the app service plan, * As Azure Web app service is a PAAS so the infrastructure that is required for hosting the Web applications get created as well   **USE CASE**   * Let's say that the app service plan created one compute instance to host two Web applications, * Now after that - we add a third Web application. If the compute instance is not enough to withstand the load of all of the three web applications. * As per the basic plan – the underlying VM can scale up to 3 instances. Hence now application can be distributed across these compute instances and the load gets distributed accordingly.   **NOTE**   * We don't have to create multiple app service plans – we can have one App service plan and have multiple Web applications as part of that app service plan. * But this decision also depends on what is the underlying Programming language for that Web application. For example, we can create an app service plan that spins up Windows based virtual machines. This Windows based virtual machine will not be able to run applications that are dependent on Linux, if we want to host Linux based web applications, then we need to create another separate app service plan. |

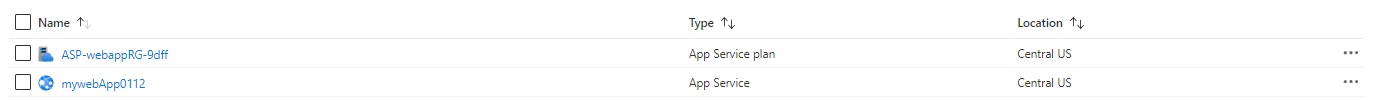
### CREATING AN APP SERVICE

* Select “Web App” Service



|  |  |
| --- | --- |
| 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 |
| OPERATING SYSTEM | The operating system that will be installed on the underlying VM which will host the webapp |
| WINDOWS PLAN | * While create a web app – an app service plan is created. * We can also to link it the web app to the existing app service plan |
| SKU / SIZE | * Here we select the type SKU/ Size available for a given app service plan |

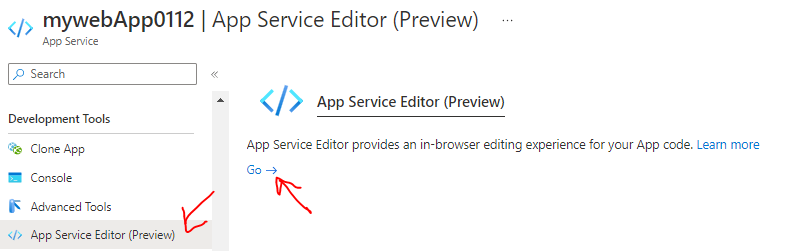
* This creates 2 resources i.e the App service and App service plan. Now if we want to create a another web app (which will use the same underlying OS ), then we can make use of same app service plan

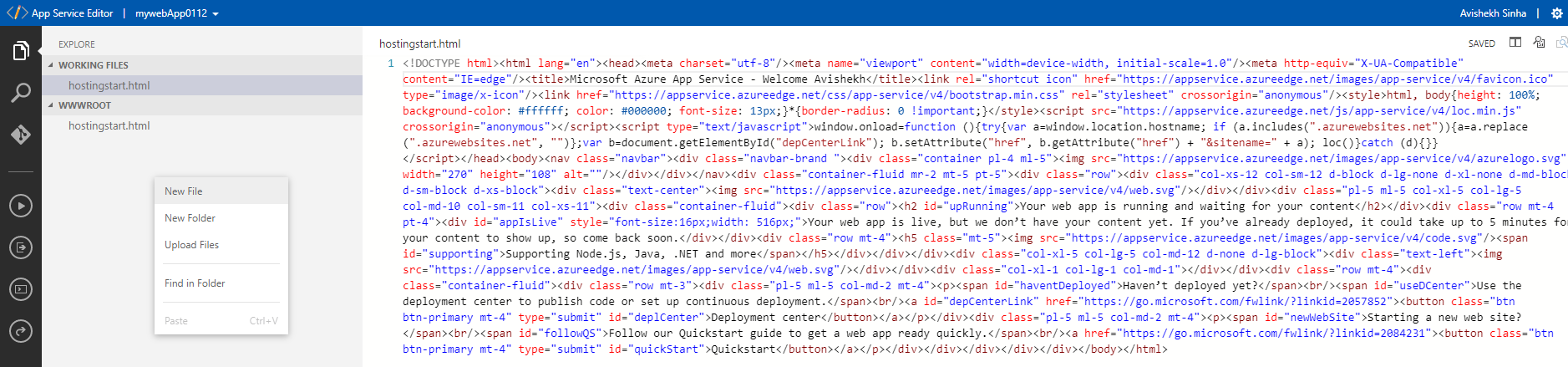


* Once the setup is done. Azure will host a default “.NET Core” based application on Azure webapp service. For our case it can be accessed at - <https://mywebapp0112.azurewebsites.net>

#### SIMPLE CHANGES IN THE WEB APP

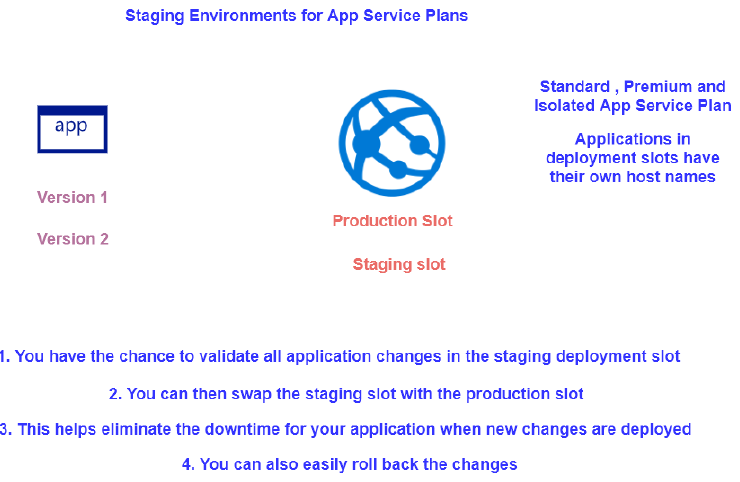
* For making small scale changes we can make use of App Service Editor. For example – updating an­­­­­­­ existing HTML, adding new HTML files etc.





### DEPLOYMENT SLOTS

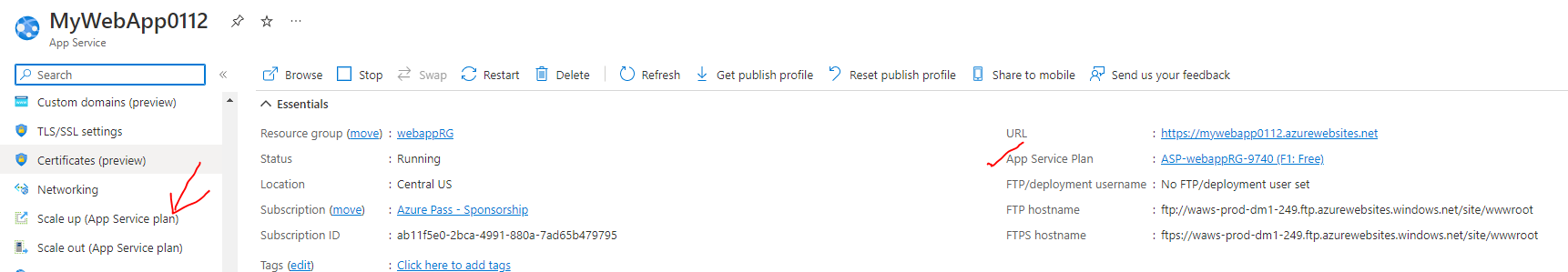
* Azure deployment slots allow us to run multiple versions of the application code on the same infrastructure.
* For example- Let's say we have a web application (*MyWebApp*) deployed in azure. By default, it gets an URL (mywebapp.azurewebsites.net). Considering it as our production version of the application then users are able to access and use it.
* Now, let's say we need to make a change to the application. (Maybe, there is a bug or adding a new feature). After the change we want to test these changes in production like environment.

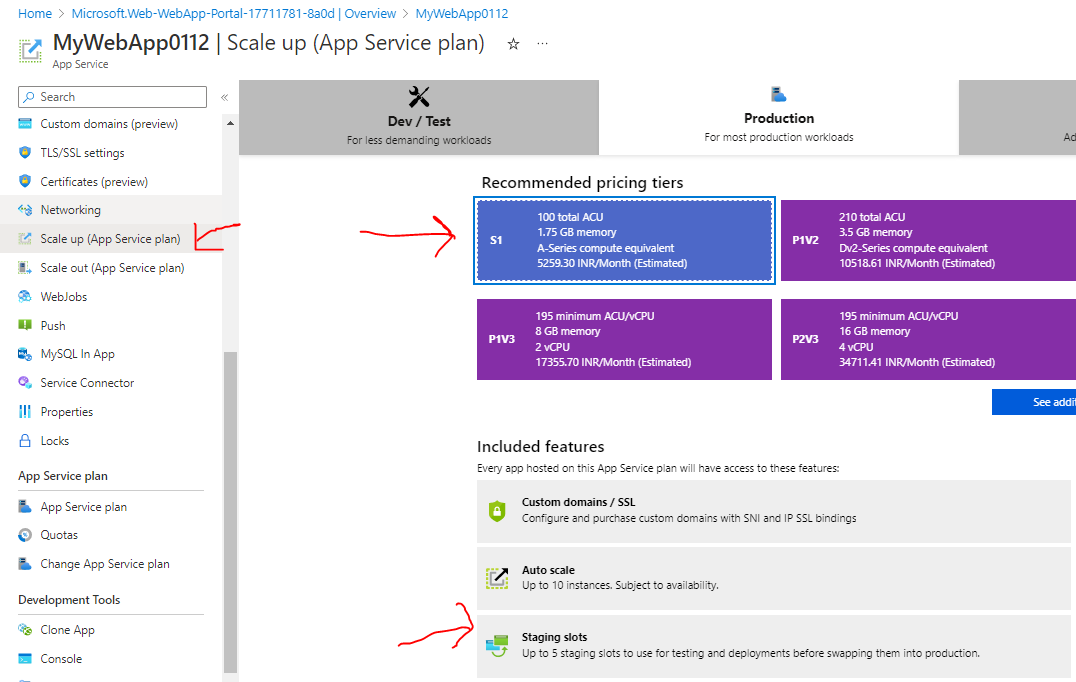


|  |  |
| --- | --- |
| azure staging slots | * We can create a deployment slot and deploy the changed version of our application. For example, if we name the deployment slot - staging, then we get the following URL. - mywebapp-staging.azurewebsites.net * The new version of the app is now in the staging slot. It gives you an opportunity to test your application. If you are happy with the changes, you simply swap the staging deployment slot with the production deployment slot. |

#### CREATING A DEPLOYMENT SLOTS

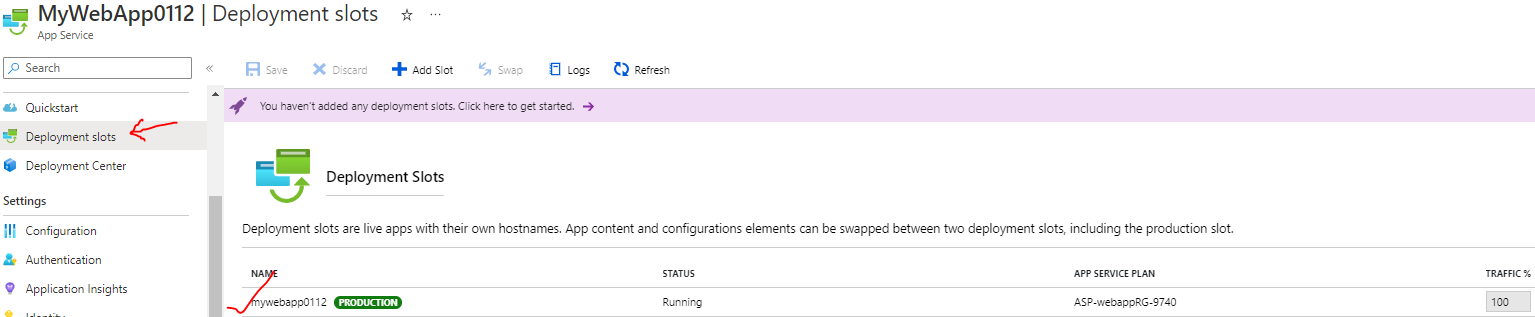
* To make use of deployment slot, the app service plan attached to the web app service should be Standard or higher. Note – we can be able to update the existing app service plan using “Scale Up” option
* Let’s say we already have a web app tied to a Free app service plan, to update the app service plan – we can make use of Scale up option.





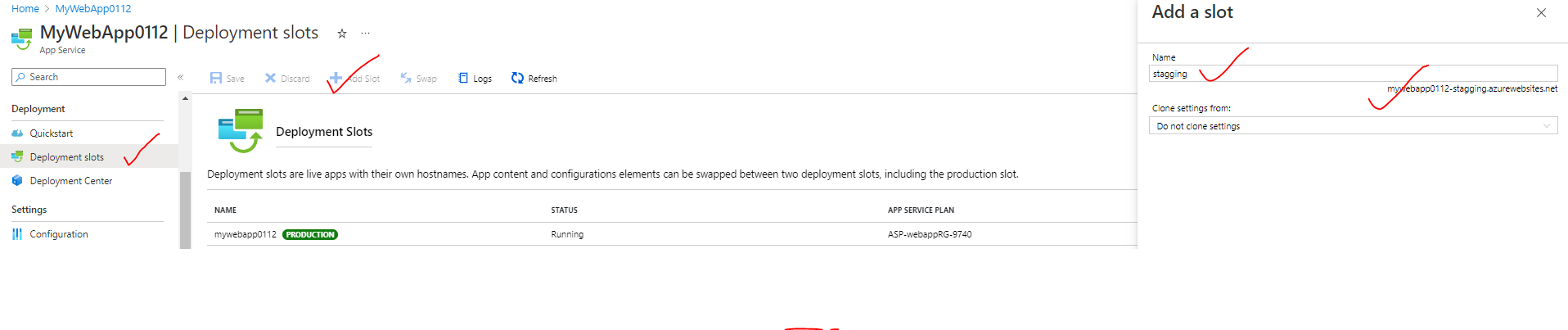
PROD VERSION OF APPLICATION

* ***PROD URL (prod version)***- <https://mywebapp0112.azurewebsites.net/>



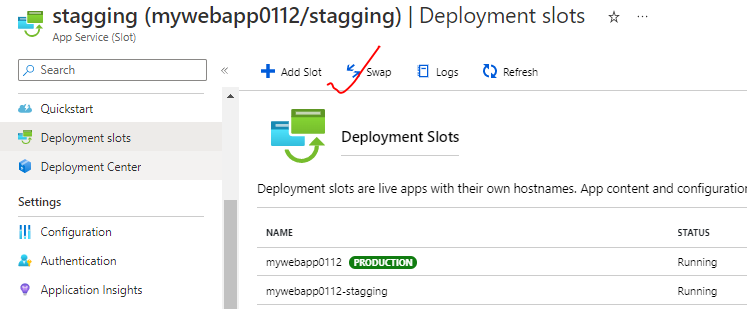
STAGE VERSION

* To deploy the newer version of the application as a new web app – In this app itself, we can create a deployment slot
* ***STAGE URL(new version)*** - <https://mywebapp0112-stagging.azurewebsites.net/>



SWAPING

* Once the testing is done of the newer version. We can now swap the deployment slots . Hence the PROD version will become STAGE version and vice versa.

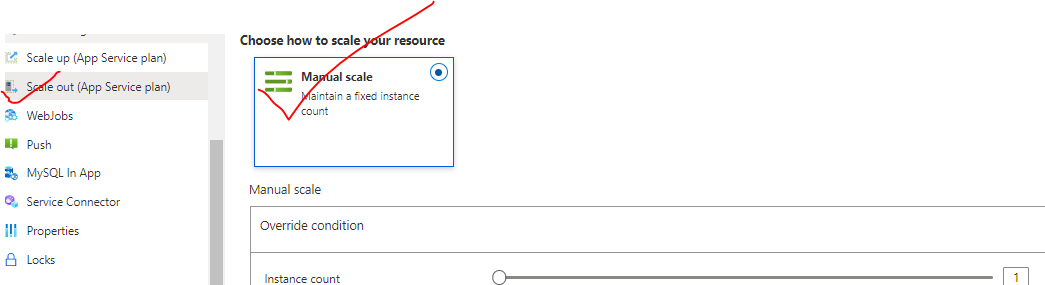


### AUTOSCALING THE AZURE WEB APP

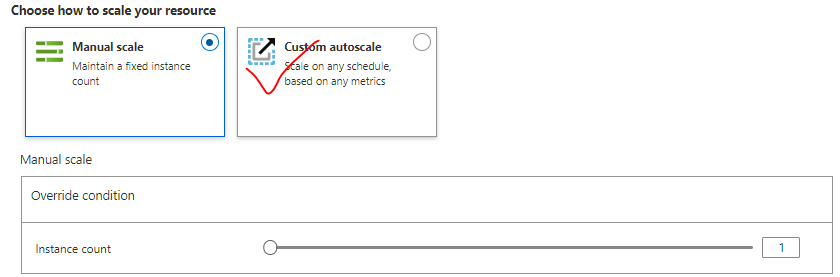
* We get the autoscaling feature if the app service plan is *Standard App Service plan or higher*.
* The autoscaling feature is useful when the traffic increases to the webapp.
* Scaling up add more compute resources which help in distributing the load across these virtual machines.
* Note –
  + The basic up service plan, we scale up to three virtual machines but it’s a manual procedure
  + For auto scaling we need to choose the standard service plan or higher.
* We need to create rules and conditions to trigger the scaling process. For example, we can create a condition based on the percentage of the underlying compute infrastructure
  + Example - if the C.P.U goes beyond a particular limit, we can scale out the number of machines automatically.
* If the load decreases, it should scale in the number of virtual machines.

#### IMPLEMENTING AUTOSCALING THE AZURE WEB APP

* For basic app plan – We can do only manual scaling

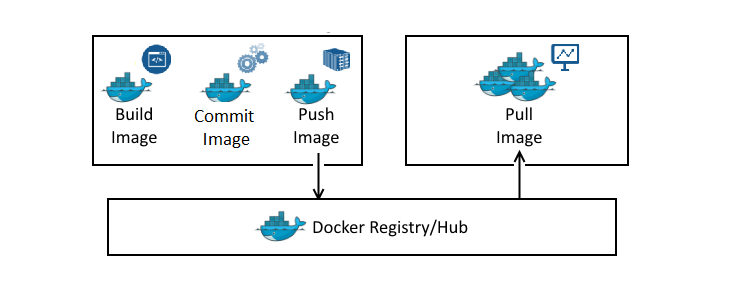


* For Standard and higher app plans – auto scaling feature is available.



##### AUTOSCALING CONDITON

## `AZURE CONTAINER REGISTRY(ACR)



* When we want to launch a container in physical / VM machine based on some image. The image gets downloaded from Docker hub, so Docker hub is a repository of images.
* On the other hand, if we create own application. We too can create a Docker image out of that application, and we want to be able to create containers out of that image. The custom image can also be uploaded in Docker hub
* **Like Docker hub, Azure has managed service called ACR. It's a private Docker registry.**

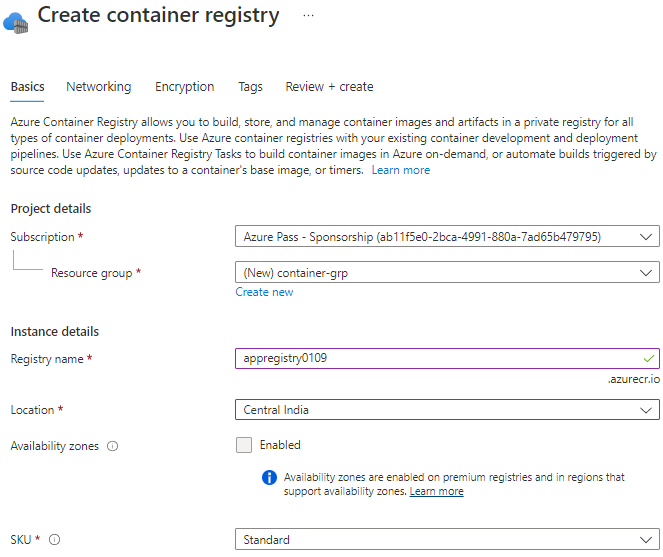


To create a container based out of the images pulled from ACR we need to go through following steps

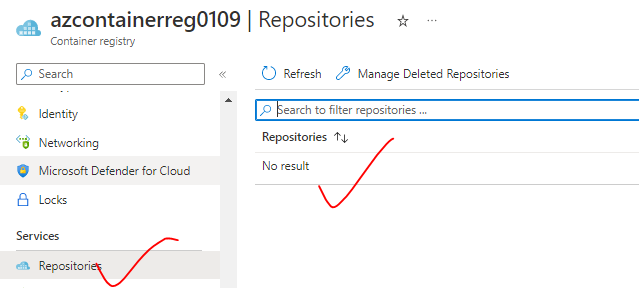
1. Create ACR using Azure’s ACR Service
2. Publish an image to the ACR
3. Create a Container Instance from Azure’s Container instance Service, then create a container based on the image in ACR.

### CREATING AN ACR

* Search for Container Registry 🡪 Create 🡪 Enter the detail as below🡪 Create



* Initially there will be no image present in the registry



### PUBLISH A DOCKER IMAGE IN ACR

1. Create a Node app and create a docker image for the same -
2. Sample Node Project - <https://github.com/avishekhsinhaRepo/Node-JS/tree/master/docker_web_app>

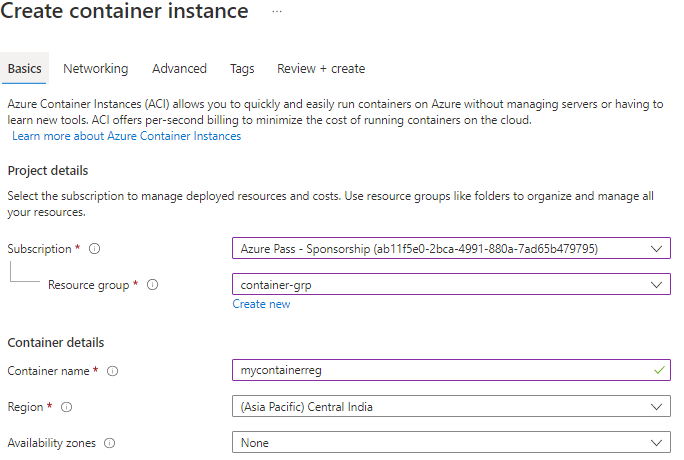
|  |  |
| --- | --- |
|  |  |
| TO BUILD THE APPLICATION | docker build . -t node-web-app |
| TO RUN THE APP IN LOCAL MACHINE | docker run -p 49160:8080 -d node-web-app |
| ACCESSING THE APP |  |

POWERSHELLL COMMAND TO PUBLISH THE IMAGE

|  |  |
| --- | --- |
| CONNECT TO AZURE | Connect-AzAccount -TenantId cf4d6b26-7de4-4fe3-8419-aab6f5713a30 |
|  | docker login azcontainerreg0109.azurecr.io |
|  | docker push azcontainerreg0109.azurecr.io/node-web-app |

### CREATING A CONTAINER INSTANCE

* Container are the instance of image. The container can be created using “Container Instance” Service.



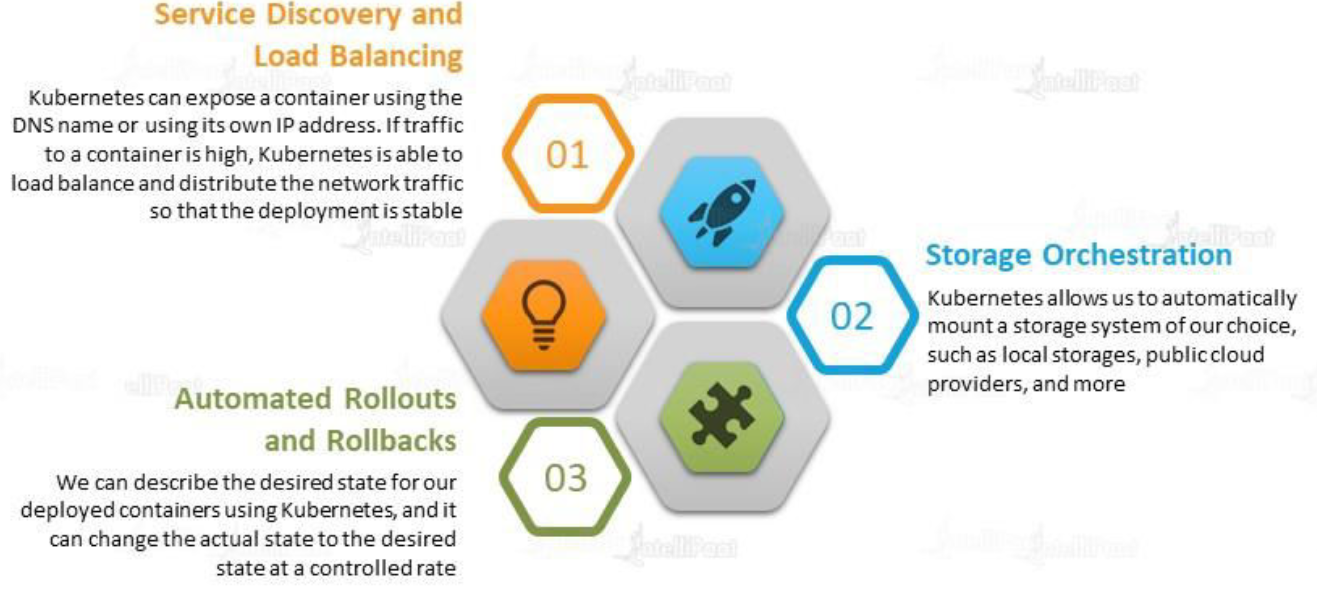
**CONTAINER CAN BE CREATED FROM MULTIPLE SOURCE OF IMAGE**

|  |  |
| --- | --- |
| IMAGE SOURCE | Option |
| MICROSOFT ACR |  |
| ACR |  |
| DOCKER HUB |  |

## AZURE KUBERNETES SERVICES

* Kubernetes is an open-source orchestration software for deploying, managing, and scaling containers
* Kubernetes manage the containers that run out application and ensure there is no downtime
* It provides us a framework to run the distributed systems resiliently.
* Azure Kubernetes is the managed version of Kubernetes tool itself

### KUBERNETES FEATURES



### KUBERNETES ARCHITECTURE

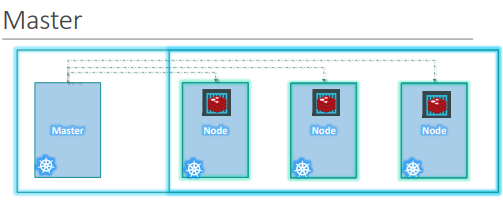
#### NODES

|  |  |
| --- | --- |
|  | * A node is a machine – physical or virtual – on which Kubernetes is installed. * A node is a worker machine, and this is where containers will be launched by Kubernetes) It was also known as Minions in the past). * But what if the node on which our application is running fails? Well, obviously our application goes down. So, you need to have more than one node. Then comes the concept of Cluster |

#### CLUSTER

|  |  |
| --- | --- |
|  | * A cluster is a set of nodes grouped together. * This way even if one node fails the application still accessible from the other nodes. * Having multiple nodes helps in sharing load as well. |

#### MASTER

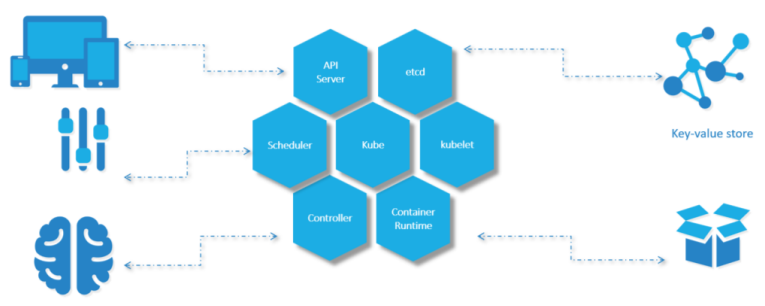


Master node is Kubernetes are

* Is responsible for managing the cluster
* Master has the information about the members of the cluster stored
* Monitoring the Nodes – For example - when a node fails it moves the workload of the failed node to another worker node

**The master is another node with Kubernetes installed in it and is configured as a Master. The master watches over the nodes in the cluster and is responsible for the actual orchestration of containers on the worker nodes.**

#### COMPONENTS OF KUBERNETES



When we install Kubernetes on a System, following components get installed

1. AN API SERVER.

* The API server acts as the front-end for Kubernetes. **The users, management devices, Command line interfaces (kubectl)all talk to the API server to interact with the Kubernetes cluster.**

1. AN ETCD SERVICE.

* ETCD key store is a distributed reliable key-value store used by Kubernetes to store all data used to manage the cluster. For example - when we have multiple nodes and multiple masters in our cluster, etcd stores all that information on all the nodes in the cluster in a distributed manner.
* ETCD is responsible for implementing locks within the cluster to ensure there are no conflicts between the Masters.
* It stores the status information of the cluster

1. A KUBELET SERVICE.

* Kubelet is the agent that runs on each node in the cluster. The agent is responsible for making sure that the containers are running on the nodes as expected.
* To manage the node - Kubelet is an agent through with the nodes are managed

1. A CONTAINER RUNTIME

* The container runtime is the underlying software that is used to run containers. For example - Docker

1. CONTROLLERS

* The controllers are the brain behind orchestration. They are responsible for noticing and responding when nodes, containers or endpoints goes down. The controllers make decisions to bring up new containers in such cases.

1. SCHEDULERS.

* The scheduler is responsible for distributing work or containers across multiple nodes. It looks for newly created containers and assigns them to Nodes.
* For example – If we fire a “kubectl” command to create new containers – it will intercepted API server and then sent over to the scheduler to schedule the task of container creation.

#### KUBERNETES IN AZURE

* Among the different components of Kubernetes – Some are Azure managed, and some are customer managed.
  + Azure Managed Kubernetes components are called “*Control Plane*”
  + Customer Managed Kubernetes components are called “Nodes”



KUBE PROXY

* This allow us to configure networking like load balancing, traffic management etc..

|  |  |
| --- | --- |
|  | * The master / worker nodes are basically VM behind the scene. * All the VMs are managed by Azure itself |

#### CREATING THE KUBERNETES SERVICE

* Search for Kubernetes service in the Azure Portals 🡪 Create

|  |  |
| --- | --- |
| NODE SIZE | Size in Underlying VM |
| NODE COUNT RANGE | Minimum and Maximum number of nodes |
|  |  |

* To deploy a container in Kubernetes cluster we can use YAML file
* The soucre of the image can be the ACR or Docker Hub

