Azure App Service

Azure App Service allows us to run applications on the cloud. Here are some features:

- HTTP based Service for hosting web applications, REST APIs, and mobile backends.
- Supports .NET, .NET Core, Java, Ruby, Node.js, PHP, Python
- Run and Scale on Windows/Linux

App Services run under an app service plan. An app service plan is the logical abstraction that represents one or more VMs that runs the app service. It consists of compute resources like CPU, memory and disk space. We pay for app service plans and not the app service.

Also, we can have more than one app service running inside an app service plan. The number of app services that can run inside an app service plan depends on the app service plan. Also, the amount of resources like CPU, RAM and disk space depends on the app service plan.

Plan	**Compute type**	**Custom Domain**	**Scaling**	**Workload**	**Space**	**Backup / Restore**	**No of Apps (max)**
Free	Shared	No	No		Nil	No	10
Shared	Shared	Yes	Yes	Dev	1GB	No	100
Basic	Dedicated	Yes	Yes	Dev/Test	10GB	No	Unlimited
Premium	Dedicated	Yes	Yes	Prod	250GB	Yes	Unlimited
Isolated	Isolated	Yes	Yes	Prod	1TB	Yes	Unlimited

Let's look at some features of App services:

Deployment Slots	This concept is used for zero downtime deployments. There will be a production slot and a Staging slot. New version of the Production deployment will be done in the Staging slot. Either all at once deployment or in stages(canary) will be done.
Deployment Center	This allows for Continuous integration/ Continuous deployment (CI / CD)
Custom Domains	By default, the website will be xxxx.azurewebsites.net. We can buy a domain in your company name and use that name.
SSL Settings	You can certificates and ensure encrypted data transmission between client and Server
Scale up (App Service Plan)	You can increase the size of your VM if you need more resources

Scale out (App Service Plan)	You can also increase the number of instances. You can either do this manually with a slider or set up rules/schedule to scale automatically on schedule or CPU usage (like >70%)

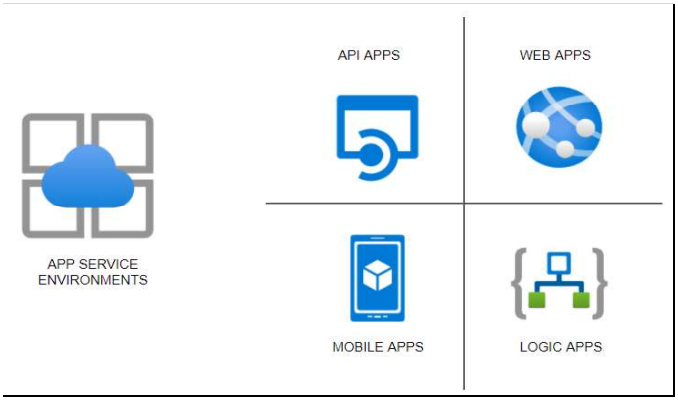
FAQs

1. **How does app service plan work?**
   App service plan is supported by Service Fabric. Service fabric replaces instances if an existing one fails. Also, it adds instances if there is a requirement.

2. **What are the types of App Services?**

   There are 4 types of services as follows:

Sl no	Type	Purpose
1	*Web App (previously Azure Websites)*	Hosting websites and web applications
2	*API App*	Used for hosting the RESTful APIs
3	*Logic App*	Used for business process automation, system integration and sharing data across clouds
4	*Mobile App (previously delivered by Azure Mobile services)*	Used for hosting mobile app back ends



App Service

- HTTP based Service for hosting web applications, REST APIs, and mobile backends.
- Supports .NET, .NET Core, Java, Ruby, Node.js, PHP, Python
- Run and Scale on Windows/Linux

Features

- **PAAS** – Patches/OS Maintenance done by Azure
- Support for Containerization and Docker
- Serverless

- **Deployments Slots** – Swap application content in Prod and avoid downtimes 
- Grouped under App Service plans with following tiers

Plan	Compute type	Custom Domain	Scaling	Workload	Space	Backup/ Restore 	Others
Free	Shared	No 	No		Nil	No	
Shared	Shared	Yes	Yes	Dev	1GB	No	
Basic	Dedicated	Yes	Yes	Dev/Test	10GB 	No	
Premium	Dedicated	Yes	Yes	Prod	250GB	Yes	
Isolated	Isolated	Yes	Yes	Prod	1TB	Yes	Private Endpoints

App Service types

1. **Webapps** – Websites/Online Apps
2. **Webapps for Containers** – Containerization
3. **API apps** – backend data

Can add – Vnet Integration / Hybrid Connections /Security, but these are not asked in the exams.

Tips

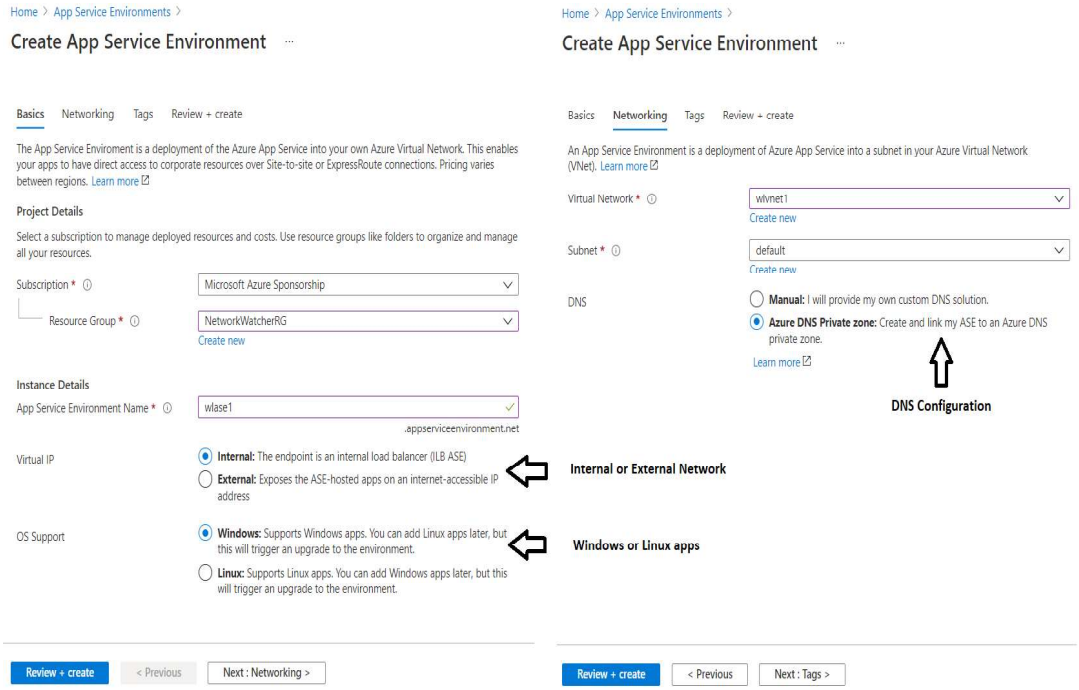
- When you move an App service from one RG to another, the App Service plan doesn't change.
- Destination RG cannot contain App Service resources like Web app or App Service plan.
- **.Net** Core application can be deployed on Windows or Linux OS
- **ASP .Net** app CANNOT be deployed on Linux OS. Only Windows OS
- Multiple Web Apps can be hosted on a single App Service plan.
- Web App and App Service plans must exist in the same region.

Application Service Environments

- There are 3 components for hosting *web apps/ Docker containers/ Mobile apps* and functions. There are app service plans which host the app services.
- When we host the regular app services, the apps are directly exposed to the internet, and the resources are shared.
- Some organizations prefer to host the services in the internal network, and security features like firewalls and security groups could be applied to protect the apps.
- For such scenarios, there is a feature called the **Azure App Service Environment**, which provides a fully isolated and dedicated environment for securely running App Service apps at a high scale.
- **App Service environments (ASEs)** provide very high scaling with isolation and secure network access with high memory utilization.
- We can create multiple ASEs within a single Azure region or across multiple Azure regions, making it ideal for horizontally scaling stateless application tiers when we have high **requests per second** (**RPS**) workloads.
  There are three types of workloads available when choosing the workload tier. They are *Dev/test, Production, and Isolated.*
- Of these, the isolated offering provides the ASE environments which host applications within the client's VNets. As stated, we have fine-grained control over inbound and outbound application network traffic.
- While the other category of app services has a fixed suffix of azurewebsties.net, we can create our own domain name.
- Also, ASEs come with powerful computers, which is twice as powerful as the regular app service plans. They also come with **1TB** Storage as compared to **50GB** of space for the regular ones.
- We can host up to 100 instances which are sufficient to host a miniature web service hub. We can expect the service to cost us about **250-300$** per month, which is very cheap for the services being provided.

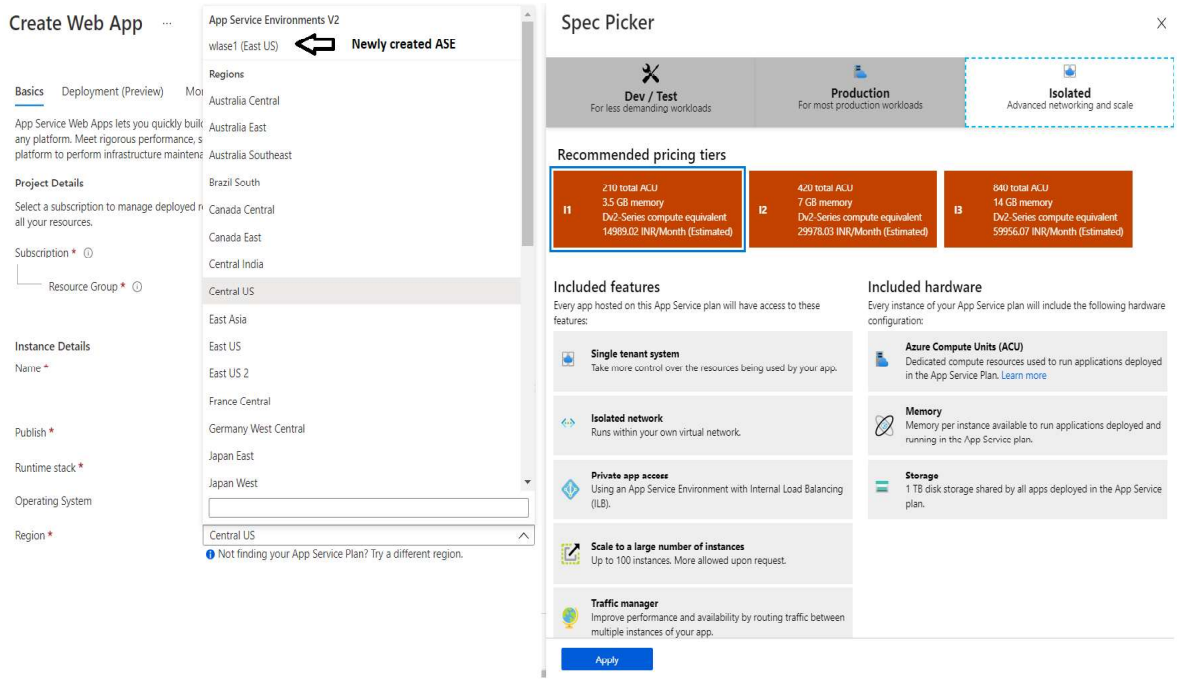
Steps to creating App Service Environment

- In the first screen, we select if the service is public-facing or internal
- Then we select whether we are hosting Windows-based or Linux-based OS.
- On the second screen, we select the Vnet where we want to host the service. *(Since services are being created in our private infrastructure, it takes much longer time to create)*
- Then we can DNS resolution. We can create our own private zone and use that name. This is not possible when choosing the other app service plans.



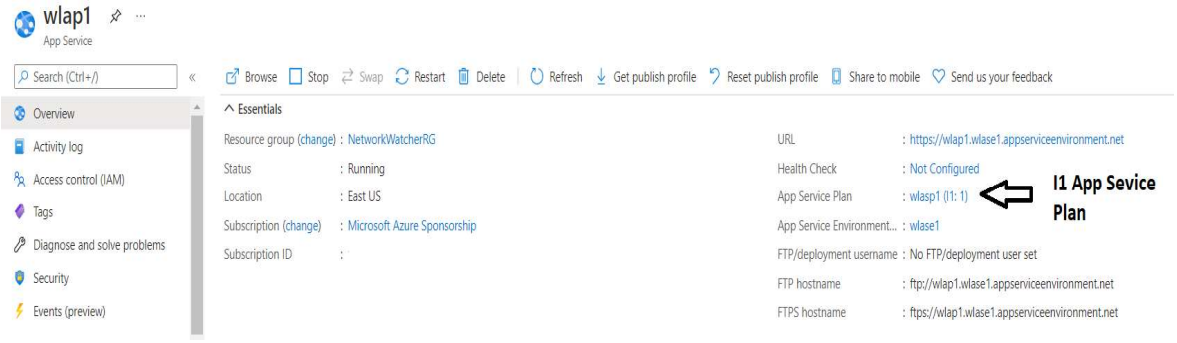
Steps to creating Web Apps under ASE

- Please note that the process is similar except that we drop down the region and select the ASE which we just created.
- Also, the below screen shows various features under ASE and pricing under each of the pricing tiers I1 and I2, and I3.



Note: *The Private link vnetLink (wlase1.appserviceenvironment.net/vnetLink) is also created below. You can go to the Resource group and click on "Show hidden types" to see this resource.*

Note: *Please see the App Service plan as I1:1 in the screenshot below to identify the isolated service plan.*



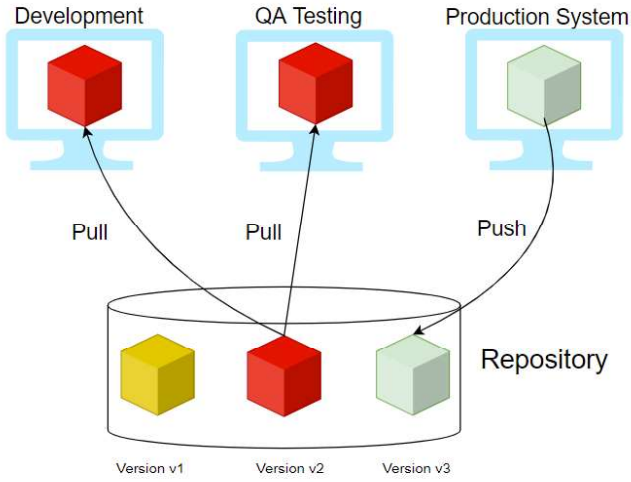
Azure Container Registry

What is a Container Registry?

A Container Registry is a central repository to store and distribute container images. A container image includes all the data needed to start a container - **for example**, the operating system, libraries, runtime environments, and the application itself.

We first build an image, and then we push the image to the repository. When needed, we pull the image into the target environment. With versioning as a feature, we have multiple versions of the container, and the different versions like the stable version would be used for Production.

Versions being tested would be in non-production regions. In the example below, v2 is a stable version, and the developer makes changes and creates v3. Once v3 is tested, it would be then pulled into Production.



Providers

Few providers provide the container registry services, and they are:

- **Docker Hub**
- **Azure ACR (Azure Container Registry)**
- **AWS ECR (Elastic Container Registry)**
- **Github Container Registry**
- **Google Container Registry**

	Amazon ECR	Docker Hub	GitHub Container Registry	Azure Container Registry (ACR)
Public Repository	No	YES	YES	No
Private Repository	Yes	YES	YES	Yes

Pricing (Public Repository)		$0	$0	$0
Pricing (Private Repository)	$	$$$	$$	
	Storage: $0.10 per GB, Data Transfer: $0.09 per GB	>= $7 per user/month	Storage: $0.25 per per GB, Outgoing Data Transfer: $0.50 per GB	Storage: $0.09 per GB
Authentication	AWS IAM	Password or Access Token	Personal Access Token (PAT)	PAT
MFA for Image Push/Pull	Yes	NO	NO	NO
SLA Availability	99.9%	N/A	N/A	99.9%
General Available	YES	YES	Beta	YES
Immutable Images	YES	NO	NO	YES
Image Scanning	YES	YES (paid plans only)	NO	YES
Regions	Choose between one of 25 regions worldwide	Not Known	Not Known	33 regions
Rate Limits	Pull: 1,000 per second, Push: 10 per second	Pull: 100/200 (Free Plan), unlimited (Paid Plan)	n/a	Pull: 1,000 per second, Push: 100 per second

ACR Service Tiers

ACR is available in 3 service tiers, also called SKUs.

1. **Basic** – Cost Optimized for developers
2. **Standard** – All features of Basic plus increased storage and image throughput. For Production
3. **Premium** – highest amount of storage and concurrent operations. It also includes geo-replication, content trust, and private link

ACR Roles

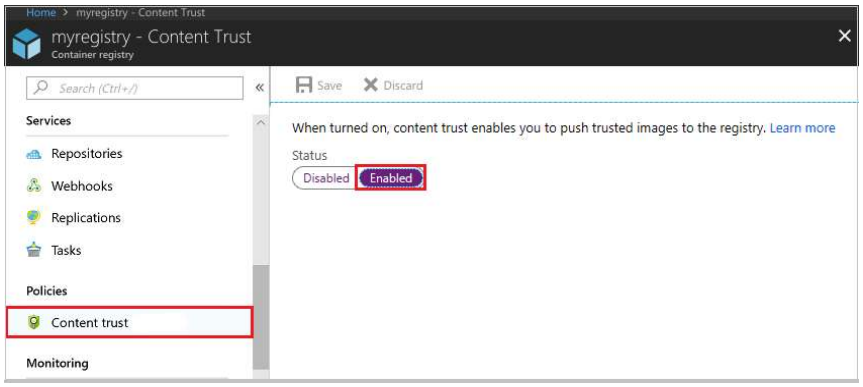
Role/Permission	Create/Delete ACR	Push	Pull	Signature Signing
Owner	X	X	X	
Contributor	X	X	X	
Reader			X	
AcrPush		X	X	
AcrPull			X	
AcrImageSigner				X

FAQs

1. **Can we change Service tiers?** –
   Yes

2. **What is geo-replication?**
   With this feature, a replica of the ACR will be created for DR purposes and local use.

3. **How can we secure the images in ACR?**
   There is a concept called CONTENT TRUST. With this, images will be signed with certificates. To enable this feature, enable registry content trust. It is available under **Policies -> Content Trust -> Enabled and then save.**



Azure Container Instance (ACI)

- Containerization is the buzzword today. Instead of spinning Physical servers and installing all the dependencies, and installing the application, we can create a container containing all the required dependencies.
- We then package and create an image and deploy it into a container.
- **Docker** is one of the platforms where we can run these containers in the Open source world. Azure has two solutions. One of those is the ACI.
- ACI is a great solution in scenarios where we need to run isolated containers. Examples are simple applications, task automation, and build jobs.
- The drawback of ACI is that it cannot be used for full orchestration like multiple containers, auto-scaling, and coordinated application upgrades. Please consider AKS for such scenarios, which is the other offering from Azure.
- In simple terms, for Production, use **AKS (Azure Kubernetes Service)**, and for simple and isolated containers, use ACI.
- One of the other best use cases for ACI is where we have production issues, and we need to troubleshoot AKS, ACI comes to our rescue where we deploy the trouble-making container in ACI and try to debug.

Advantages of ACI

- *Fast Startup times*
- *Container access*
- *Custom Sizes*
- *Persistent Storage* – We do this by mounting Azure file shares.
- *Virtual Network deployment* – When deployed in a Vnet, ACI can securely communicate with other resources in the Vnet.

FAQs

1. **What are probes in ACI?**
    - You can configure the liveness probe. We check the liveness probe to see if the container is healthy. If the container is not healthy, we need to restart. There are common scenarios when containers run for a long time.
    - You can configure the readiness probe. Here we might have a scenario where the container (maybe DB for the backend) is just coming up. We run the readiness probe and send requests to the container only if the probe succeeds.

2. **How can we monitor ACI?**
   We use Azure Monitor. Here are the available metrics at this time.
    - CPU Usage measured in millicuries (One millicore is 1/1000th of a CPU core)
    - Memory Usage in bytes
    - Network bytes received per second.
    - Network bytes transmitted per second

3. **What are container groups?**
    - Similar to AKS for orchestration, we can use container groups to combine and manage containers. They get scheduled on the same host machine.
    - The concept is similar to pods in Kubernetes. The use case for this is in scenarios where we want to divide a single functional task into a smaller number of container images. An example is a front-end container and a back-end container.
    - The front end might serve a web application, with the back end running a service to retrieve data.

Azure Kubernetes Service (AKS)

What is Containerization?

- In the traditional computing system, we had to install an Operating system and install all dependencies for an application to work. Only a single OS could be installed.
- Then came Virtualization where we could install multiple OS by introducing another layer between the hardware and the OS and this was called Virtualization. So only physical machines appeared as multiple systems.
- Then came a lightweight alternative to virtualization, which was called Containerization. This removed the drawback of having a full machine, and this had only the necessary components.
- Containers will encapsulate an application with its operating system. This would contain all the dependencies that were needed for an application to run. So we take the container and run it on any operating system, and it will run.
- Some of the containerization options are Docker, which is the most popular and sometimes equated to containers. But there are others like **LXC/LXD, ContainerD, Rocket**.

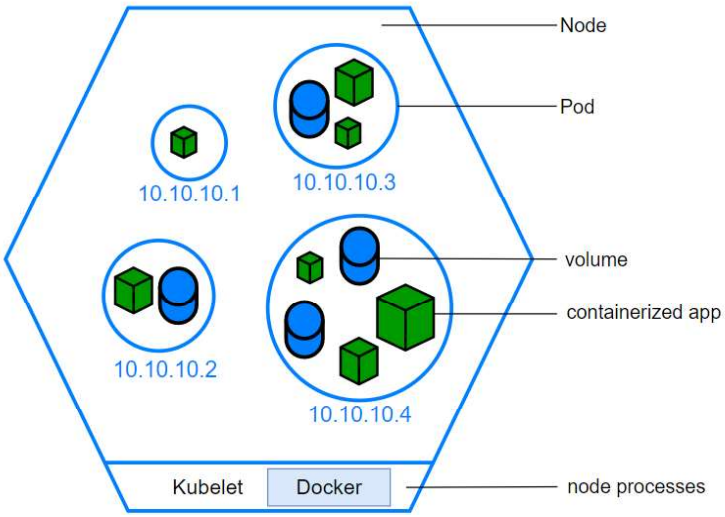
Orchestration

- Orchestration is the system that is used to manage the deployment of containers. We use Orchestrators as tools to achieve this. Some of the performed activities are automating the maintenance of those applications, replacing failed containers automatically, and managing the rollout of updates and reconfigurations of those containers during their lifecycle.
- The popular tools are *Docker Swarm by Docker, Nomad by Hashicorp, Flocker, and Kubernetes.*
- Kubernetes, also stylized as K8s, is an open-source container orchestration system. It is used for automating computer application deployment, scaling, and management. It was originally designed by Google and influenced by Google's Borg System and is now maintained by the Cloud Native Computing Foundation. It is a cluster management software for Docker containers mainly but supports others also.

AKS

Kubernetes has become very popular, and many cloud service providers offer a Kubernetes-based platform or infrastructure as a *PaaS or IaaS offering. Google has GKE (Google Kubernetes Engine), AWS has EKS (Elastic Kubernetes Service), and Azure has AKS (Azure Kubernetes Service)*

Components of AKS



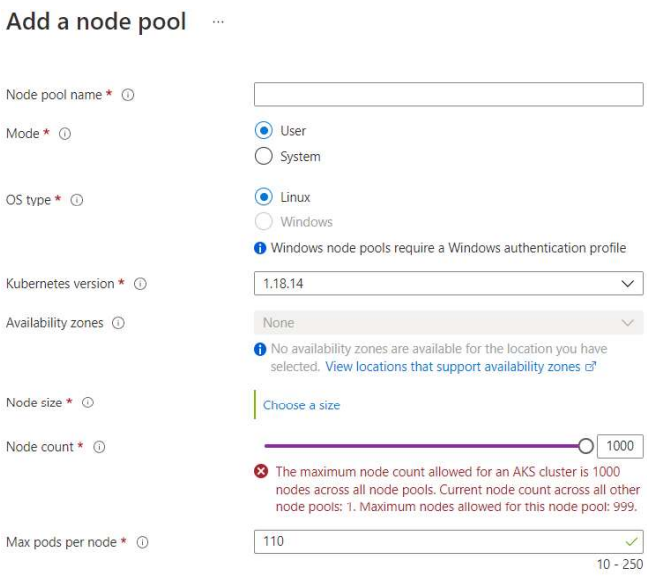
1. **The Cluster**
   - The Cluster contains 2 components
     - Control Plane – this consists of kube-apiserver, etcd, kube-scheduler and kube-controller-manager
     - Nodes that run the applications
2. **Persistent Volumes**
   - Since the nodes are added and removed on-demand and the storage associated with it is temporary, we need to create storage outside of the cluster. Hence we create persistent volumes.
3. **Node**
   - We create Node pools in Kubernetes (as shown below). Here we choose a VM size, and that will be the unit size of the nodes within the pool.
   - We can add node pools as needed. The first node pool created is the **system node** pool which hosts critical system pods like coreDNS and tunnel front.
   - We then add user node pools for application support and create different pools based on the application requirements.
   - Pods will be created within the nodes, and the max pod setting is configured at the node pool level.

Node pools

+ Add node pool | Refresh | Delete | Upgrade | Scale

You can add node pools of different types to your cluster to handle a variety of workloads, scale and upgrade your existing node pools, or delete node pools that you no longer need.
Learn more about multiple node pools

Name	Mode	Provisioning state	Kubernetes version	Availability zones	OS type	Node count	Node size	Max pods / node
default	System	Succeeded	1.18.14	None	Linux	1	Standard_D2_v2	110



4. Containers

- We store our code that is going to be run inside containers. There are readily available pre-built containers stored in container repositories or we can create our own containers.
- One or more programs can be run from the containers

5. Pods

- Nodes create Pods, and kubernetes use Pods to run instances. Usually, only one container is run within a pod, but multiple containers could run in a pod if there was a requirement from the application.
- We scale based on pods. When we can scale, we simply use pod replicas. A new pod will be spun up in another node, and we now have an additional pod. Same way, we can remove the pods to scale down.

6. Deployments

- We don't launch pods directly. Instead, we create deployments.
- A deployment will state how many replicas should run and the system manages that.

Sample Deployment yaml file

```
apiVersion : apps/v1
kind: Deployment
metadata:
  name: wl-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: wl-app
  template:
    metadata:
      labels:
        app: wl-app
    spec:
      containers:
        - name: wl-app
          image: wl66293099.azurecr.io/wl-app
          ports:
          - containerPort: 3000
```

7. Ingress

- By default, Kubernetes provides isolation between pods and the outside world. If you want to communicate with the service running in the pods, you need to open the communication. This is called Ingress.
- You can achieve this communication in several ways. The most common ways are Ingress controller or a load balancer. Please see the sample service.yaml file which creates an external load balancer. We get the IP of this service and connect.

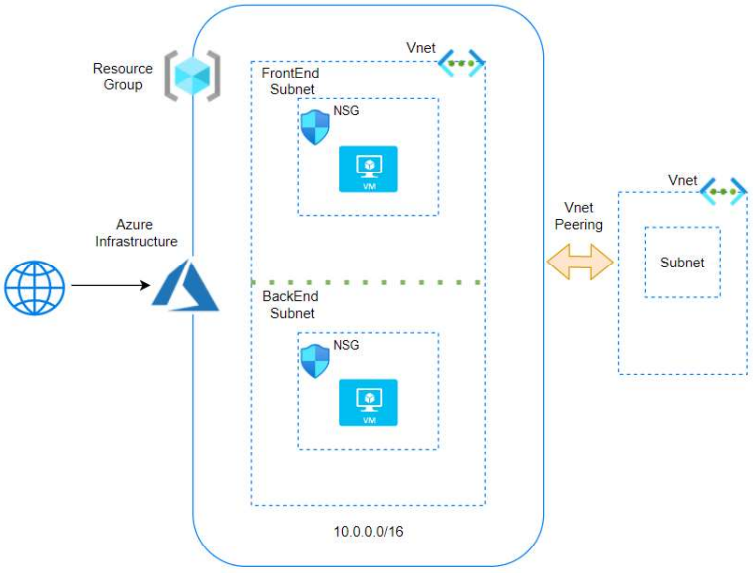
```
apiVersion: v1
kind: Service
metadata:
    service.beta.kubernetes.io
    name: wl-app
spec:
    type: LoadBalancer
    ports:
    - port: 3000
    selector:
        app: wl-app
```

Azure Virtual Network

A **Vnet** is the fundamental building block for a private network in Azure. Vnets allow azure resources like VMs to communicate securely to each other, to the internet and on-premises networks. A **Vnet** is the representation of our own network in the cloud. We can logically isolate resources within our Vnet.

Benefits of Vnet

- **Isolation** – As discussed, the components of a Vnet are isolated. We can connect to other Vnets or On Premises with Vnet Peering or VPN or Express route
- Access to the public network
- Access to VMs within the Vnet
- **Name resolution** – We can resolve to other components in the Vnet and address them
- **Security** – We can secure the components at various levels in the Vnet
- Connectivity



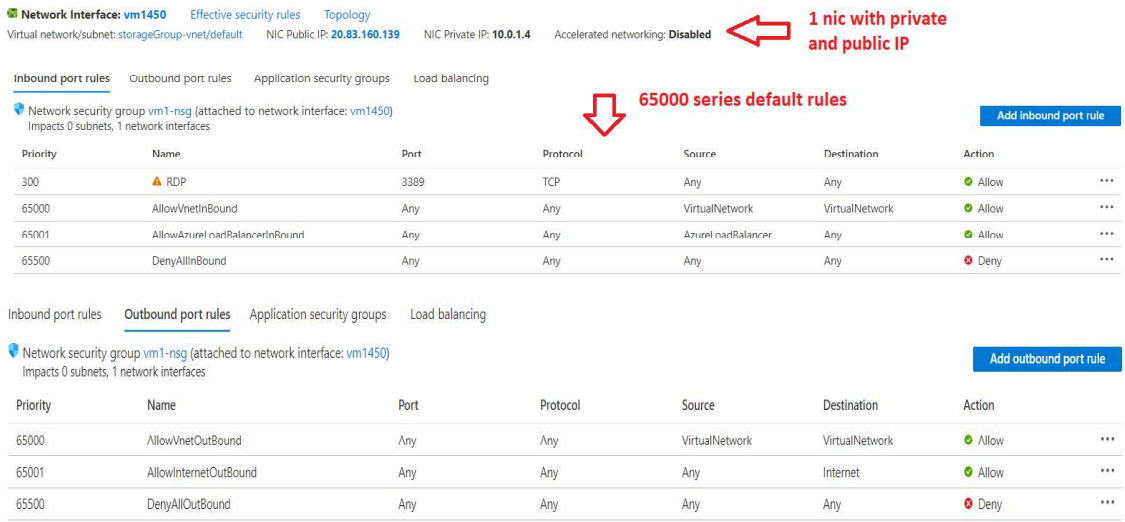
Components of Vnets

- **IP addresses**
  - **Public and private IP addresses**
    - The Vnets are configured with a range of IP addresses. The Notation is in CIDR.
    - By default, Private IP addresses are assigned to the resources with which communication takes place between the resources
    - Optionally, Public IP address can be assigned to the resources
    - Please note that we will pay for Public IPs if they are not assigned. This is to conserve Public Ips

- **Subnets**
  - A Subnet is a subcomponent of Vnet. All resources must exist in a subnet. A default subnet is created when a Vnet is created.
  - Access can be restricted at a subnet level also
  - Let's say we have 2 tiers in an application called Front end and Back end. We can create 2 subnets and configure access in such a way that internet traffic will flow to the front end subnet and from there to the back end subnet.

- **NIC - Network interface card**
  - A NIC is the networking component which allows traffic flow. A single NIC will contain the public and private address.

- **NSG - Network security group**
  - These are the rules that are assigned to allow traffic to flow. The NSG can be assigned at a NIC level or a subnet level. It is recommended to apply at any one level only.
  - If there is no NSG, then traffic will be allowed in and out
  - We set inbound and outbound rules
  - **Priority** – All rules are assigned a priority and the lowest number is taken first. If rule 100 says allow and 101 says deny, then the result is allow.
  - **Default Security rules** – There are 6 default rules that can neither be removed or modified.



FAQs

1) **What is Vnet Peering?**

   Vnet Peering allows two Vnets either in the same region *(Default Vnet Peering) or Globally (Global Vnet Peering)*

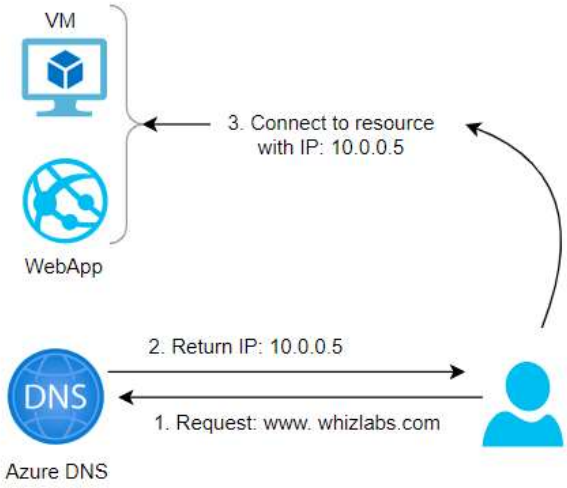
2) **What are the pre-checks for Vnet Peering?**
    a. Peering is non-transitive. If Vnet A is peered with Vnet B and Vnet B with peered with Vnet C then it does not mean that Vnet A and Vnet C are connected
    b. The Address ranges cannot overlap between the Vnets
    c. When peered, adding or deleting address range is disabled. If we need to add address range, we need to delete the peering and add the address range and then add peering again.

Azure DNS

What is DNS?

Think of the phone directory that is used at home. It is difficult to remember a string of numbers and hence the phone directory will list the phone numbers with names of persons/businesses.

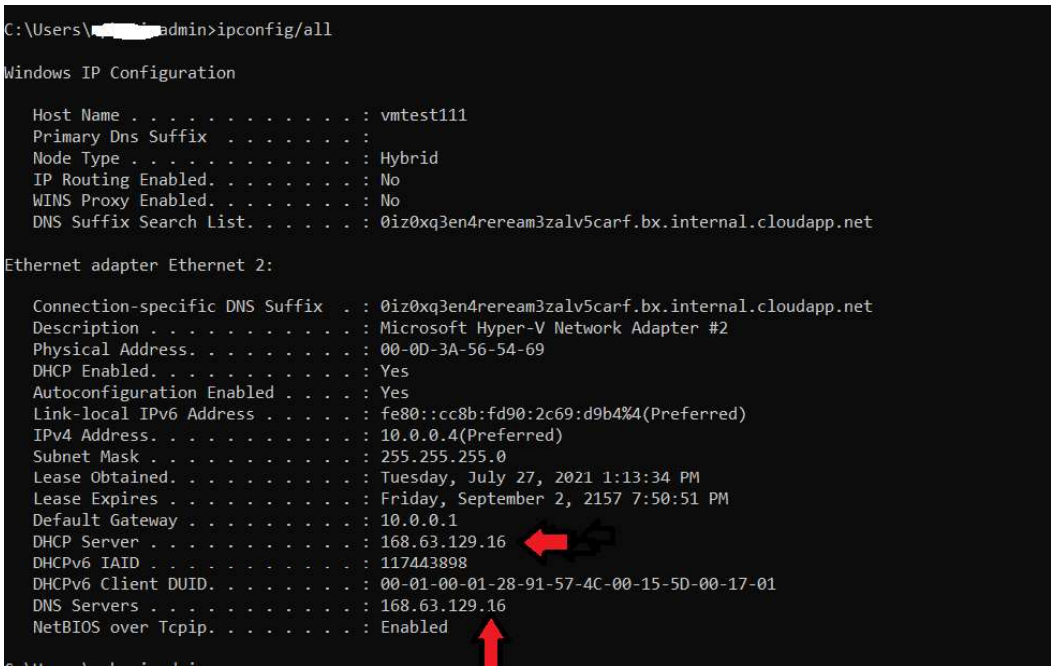
- Coming back to the IT world, computers communicate with IP addresses. The DNS (Domain naming system) is a friendly name given to the computer.
- For example, a web server has an IP address of **53.102.94.86.** Instead of using the IP Address, we assign a host name as web1. In a domain, the **FQDN (Fully qualified domain name) will be web1.demystify.com.**
- This is facilitated by DNS Servers which are setup in a hierarchy. At the top most level, we have the ROOT and under the root, we have the top level domains (TLD) examples of which are **.ORG, .COM, .NET, .IN etc.,**
- In addition to this, we have domain registrars where we purchase a domain name.
- Examples are **Godaddy, Namecheap and Amazon too via Route53**. When a user tries to connect to a server demystify.com, the DNS resolves this to the IP address by going to the **ROOT** and then to the **.COM server.**



- DNS works with a concept of Zones. We can set up Private or Public zones. Public zones are used when we want the internet to be able to resolve our names.
- However when we want to enable internal communication, we create private zones.
- Please note that zones can also be configured with a "**Split-horizon**" view which allows a private and public DNS zone to share a name.

FAQ

1) **What is IP 168.63.129.16?**
   This is actually called a Wire Server and has an IP address of 168.63.129.16. and it facilitates communication between Azure resources. It also serves as a DNS and DHCP server by default. Please ensure that this IP is not blocked.



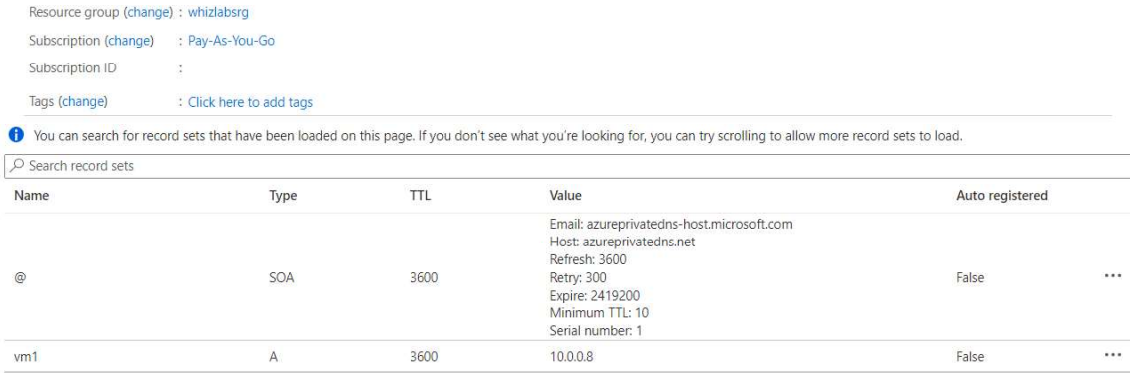
Connection-specific DNS Suffix . : lhlv032okq5e3g5zezobyk5bwf.bx.internal.cloudapp.net

Description . . . . . : Microsoft Hyper-V Network Adapter #2
Physical Address. . . . . : 00-0D-3A-8E-15-4C
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::7dbd:c33b:1ab:8e7f%7(Preferred)
IPv4 Address. . . . . : 10.0.1.4(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Saturday, March 13, 2021 7:06:42 PM
Lease Expires . . . . . : Wednesday, April 20, 2157 9:38:31 AM
Default Gateway . . . . . : 10.0.1.1
*DHCP Server . . . . . : **168.63.129.16***
DHCPv6 IAID . . . . . : 117443898
DHCPv6 Client DUID. . . . . : 00-01-00-01-27-DE-C5-9A-00-15-5D-00-04-01
*DNS Servers . . . . . : **168.63.129.16***
NetBIOS over Tcpip. . . . . : Enabled

2) **Can I buy my domain from Azure?**
   No, Azure is not a domain registrar. You need to buy from a domain registrar and you can create a zone in azure and add the records for DNS resolution.
3) **How do we configure VMs to use private zones?**
   We can configure auto registration and for Vnet that we link with the Virtual Network Link on the DNS Zone, the DNS registration will be done automatically when the VM is created.

4) **How do I use my custom website?**

We need to create a public zone and add an alias record. Once verified with the registrar, we can start using our custom name.



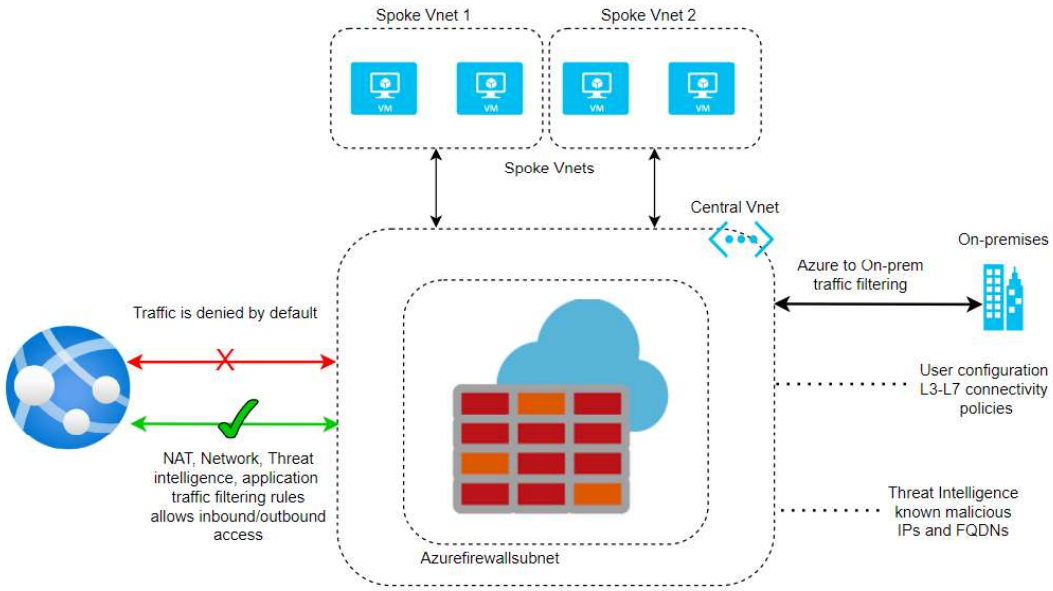
Azure Firewall

What is a Firewall?

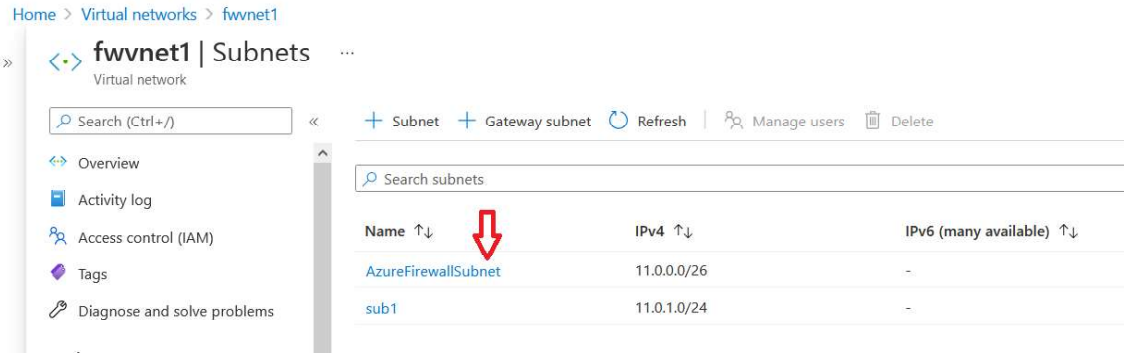
A Firewall is a security device for the network that monitors both incoming and outgoing traffic. Based on a set of security rules, it will either allow or deny the traffic. It acts as a barrier between our network and traffic from external sources like the internet. The objective is to block malicious traffic which include hackers and viruses.

Azure Firewall

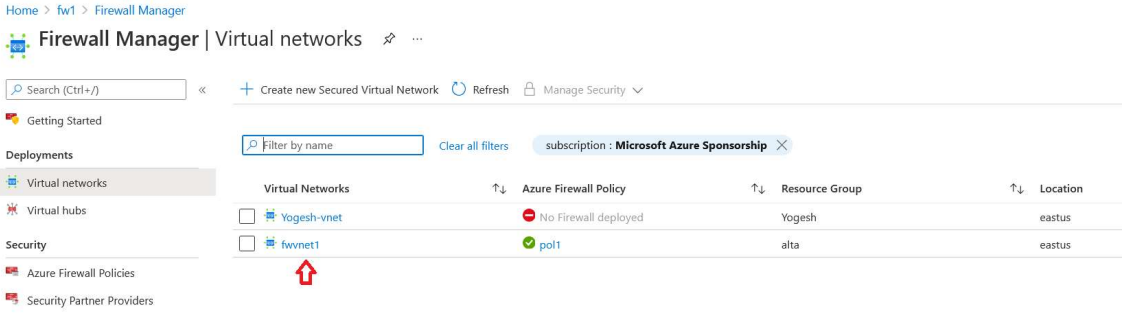
- Azure Firewall is a **network virtual appliance (NVA)** which is a managed network security device on the cloud.
- The function is to protect our network resources on the cloud. There are two types of firewalls and they are classified as either Stateful or Stateless. Let's say that you allow a certain incoming traffic *(say port 80).*
- When the same traffic returns, it is automatically allowed if it is stateless. On the other hand, Stateful traffic will need a specific rule for the outgoing traffic also, else the traffic will be blocked.
- Azure Firewall is a fully stateful firewall. So, we need to allow both incoming as well as outgoing traffic.
- Azure Firewall has built-in high availability and is highly scalable. We can create, enforce, and log application and network connectivity policies across subscriptions and virtual networks from a central location called **Firewall Manager.**
- We need to set up a static public IP address for the virtual network resources allowing outside firewalls to identify traffic originating from the virtual network. It is fully integrated with **Azure Monitor** for logging and analytics.
- A typical setup for the firewall is done via a hub and spoke model where the Vnet which hosts the firewall will act as a hub and the other Vnets will act as a spoke.
- The On premises and Internet is also connected to **Azure Firewall.** In this way, all traffic will enter via the firewall and the rules setup via the policies will then allow or deny the traffic.
- Please note the subnet that hosts the firewall must be named as Azurefirewallsubnet else it will not function



- Please see below the subnet created for the Azure firewall named as **AzureFirewallSubnet.**

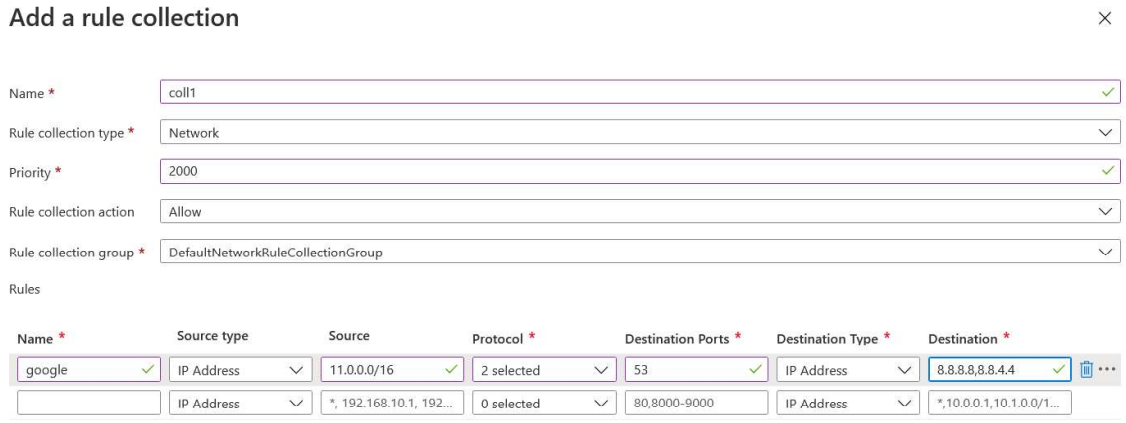


- As discussed, the rules are set up in a central location using the Firewall Manager. You can see the pol1 being assigned to **fwvnet1 Virtual Networks**. We can assign the same policy to other networks and it is easier to manage centrally.

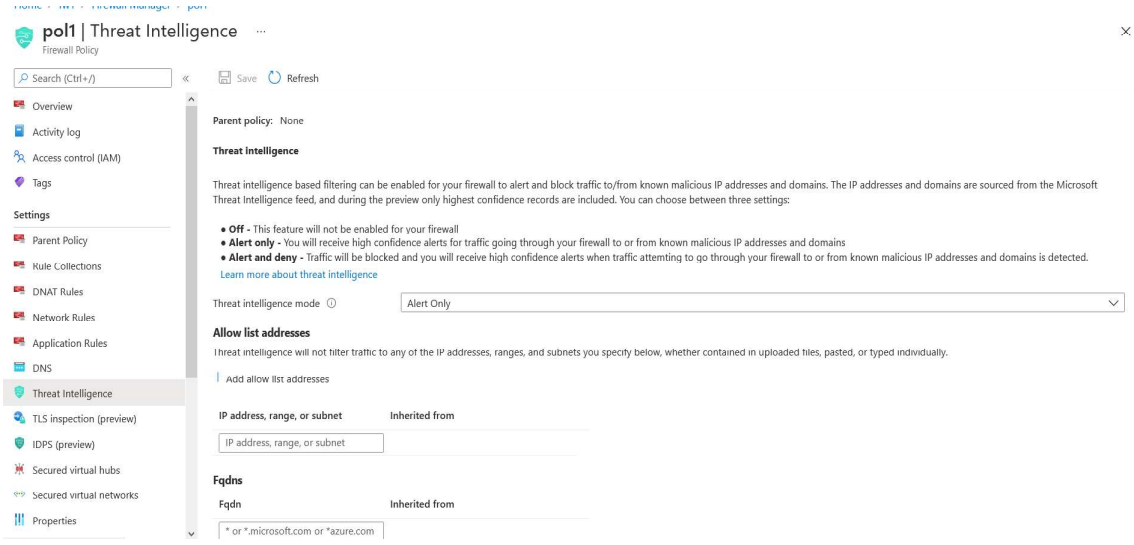


- A Policy consists of rule collections which in turn contains individual rules. Here we specify if the rule is to allow or deny.

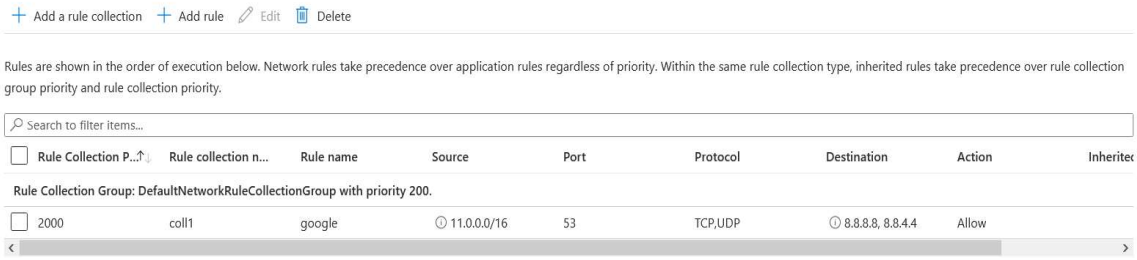
- We assign a priority from **0 to 65535** and the lowest number takes the priority while processing the rules.
- We could place the rule collection within a group called the rule collection group. Also, the rule is available as a tab called Network rules on the main panel.
- We specify the source type as either an IP address or IP Group. We can give a range of IP addresses for Source and Destination. We can give ***** to indicate all.
- We can specify Protocol and Port numbers. In the example below, we have given Google a DNS server with IP of **8.8.8.8** and port of **53** which will allow DNS resolution.



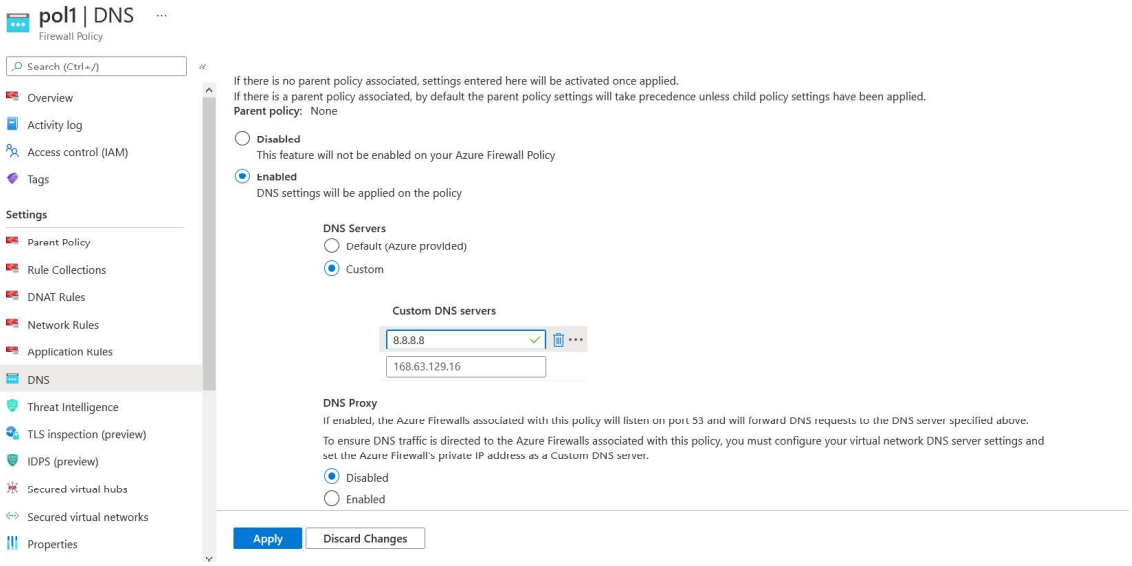
- We can optionally enable intelligence-based filtering called Threat Intelligence and the mode can be set to OFF/Alert only or Alert and deny. Microsoft threat intelligence feed provides a list of IP addresses and domains and these recorded are included as rules to allow or deny



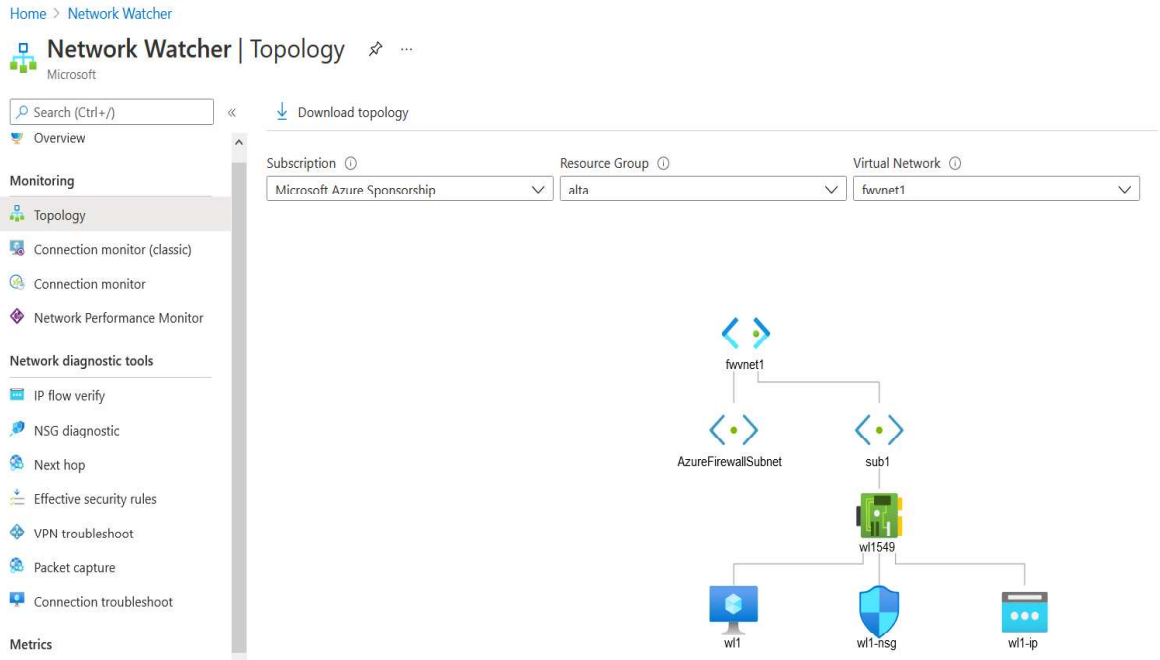
This is the Network Rule tab which lists the rules.



We can also set up DNS servers for DNS resolution on the DNS tab.

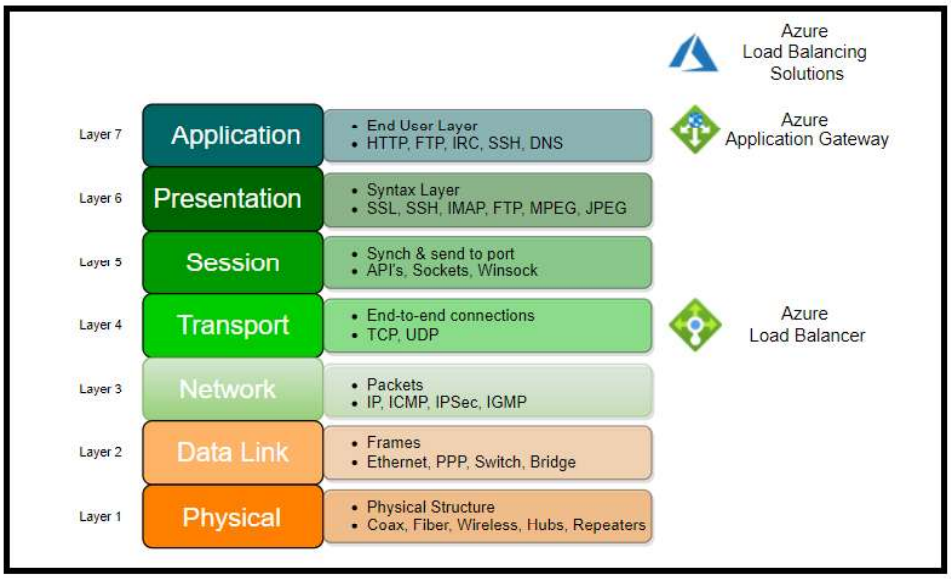


Finally, we can see the topology of the Vnet and the firewall subnet on the Network watcher blade under the Topology tab.

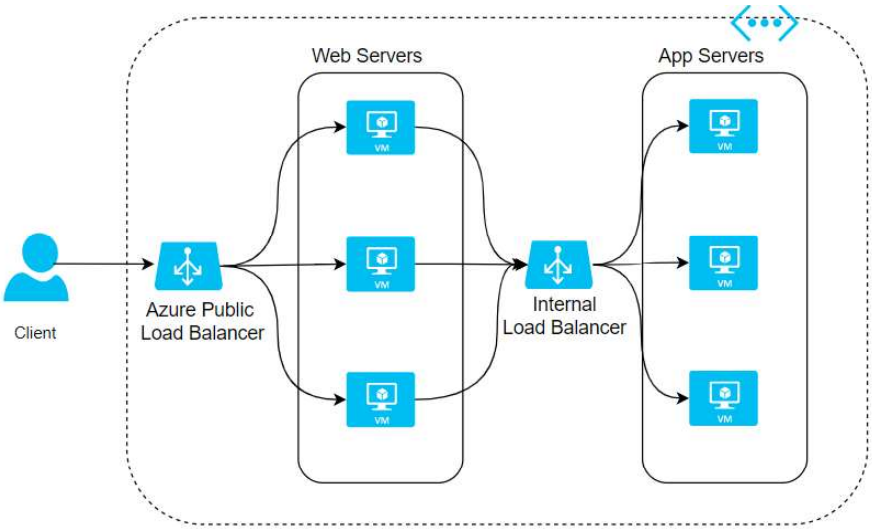


Azure Load Balancer

- Azure provides load balancing at **Layer 7** which is the application layer via Azure Application Gateway. This is typically http traffic.



- Azure also provides load balancing at Layer 4 which is a transport layer consisting of **TCP and UDP** protocols. This is the Azure Load Balancer.
- We could use the Azure Load balancer for both public facing as well as internal application. The load balancer is set up with a backend pool which distributes traffic to a set of VMs or VM Scale sets.



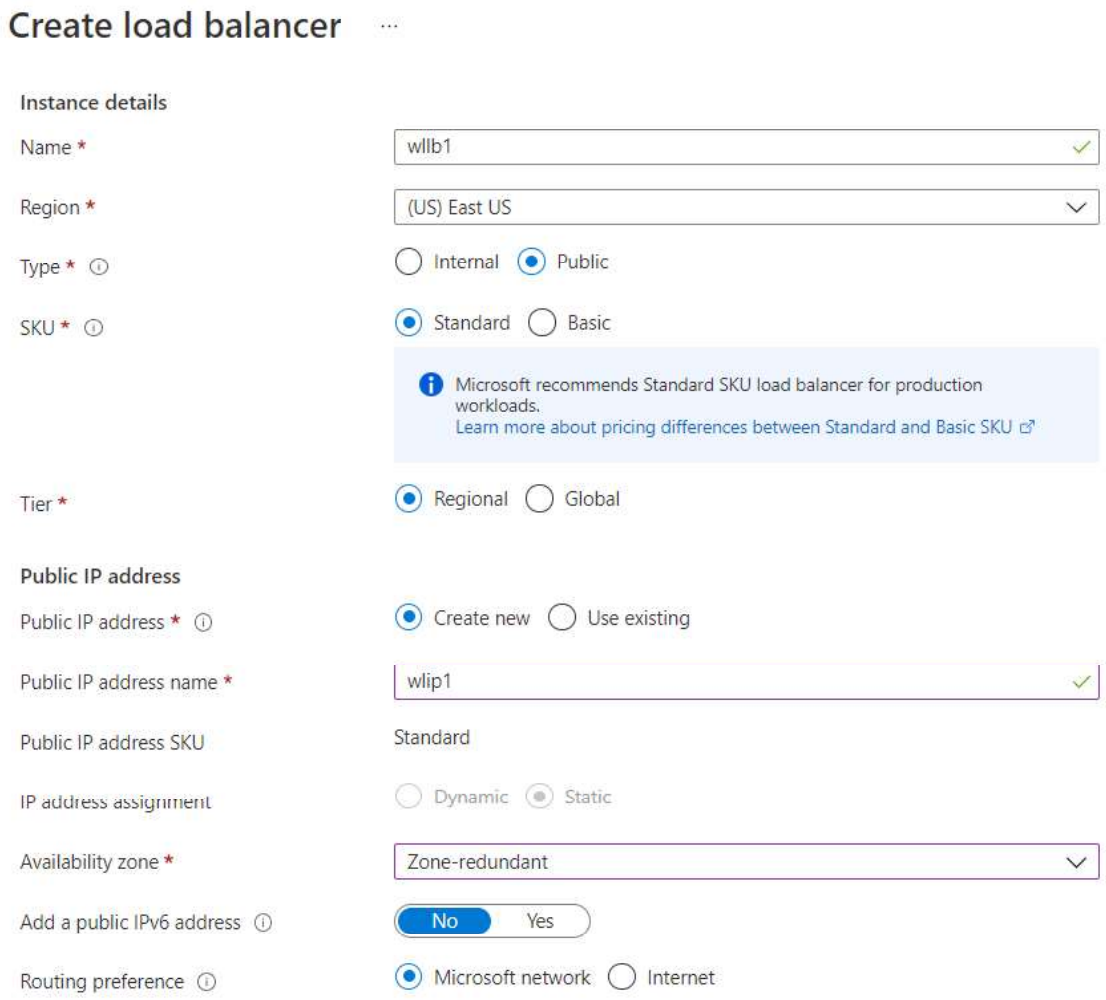
Here are the steps to create a load balancer:

Step 1: Create Load Balancer

We create a load balancer with the following options:

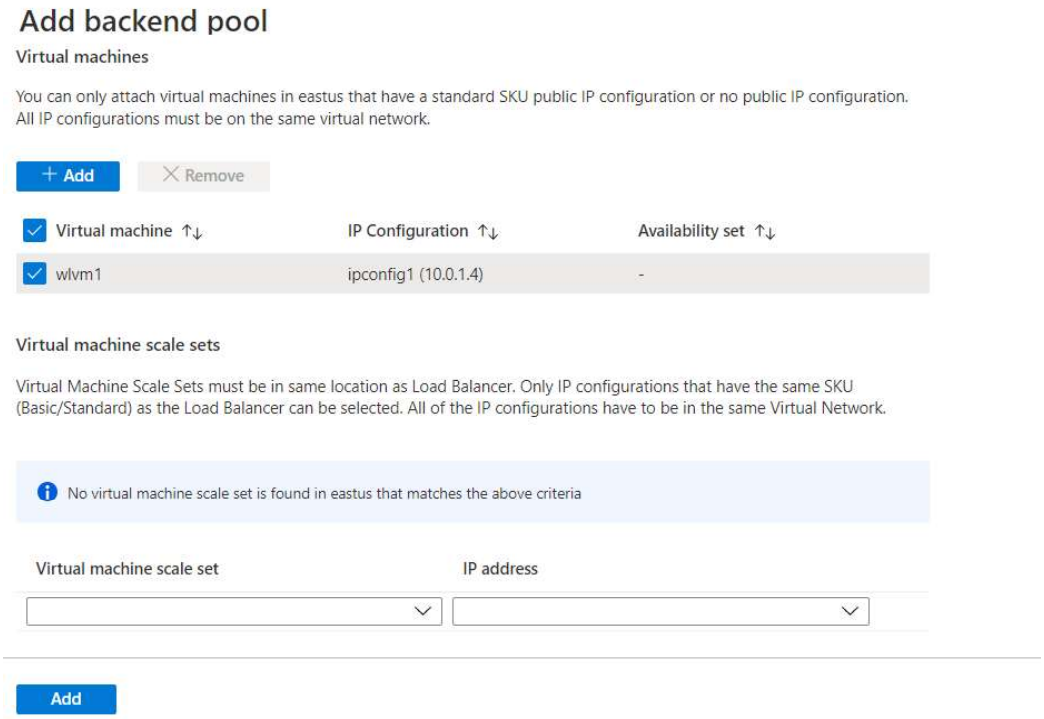
- Name for the load balancer

- Internal or Public load balancing
- SKU type could be Standard or Basic. Since Basic does not have an SLA, Standard SKU type is recommended for Production workload which has SLA of **99.99%.** *Standard SKU* comes with many more additional / better features than Basic SKU like https.
- **Regional or Global** – This is a new feature and is available for Public Load balancers



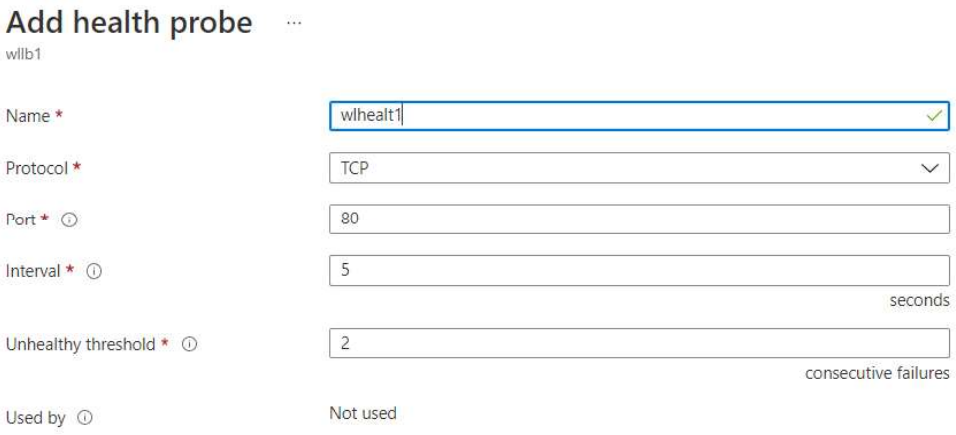
Step 2: Create Backend pool

- We create a backend pool where we attach VMs or VMSS
- VMs/ VMSS have to be in the same location.
- We could add multiple backend pools



Step 3: Add Health Probe

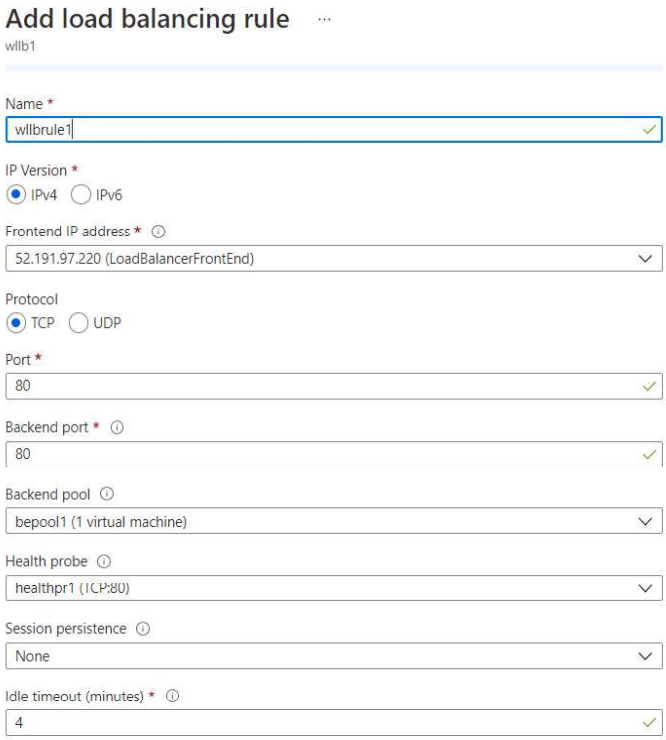
- We need to add a health probe
- We can configure **TCP/HTTP/HTTPS** as protocol
- We add a port number
- We add an interval and unhealthy threshold which is the interval for checking where the probe passes a health check. The unhealthy threshold is the number of times a probe is allowed to fail consecutively after which the instance will be marked as unhealthy and traffic routing will be stopped.



Step 4: Add Load Balancing rule

- We create a load balancing rule
- We specify frontend IP address and Protocol (TCP or UDP) and Port
- We specify the Backend port and pool

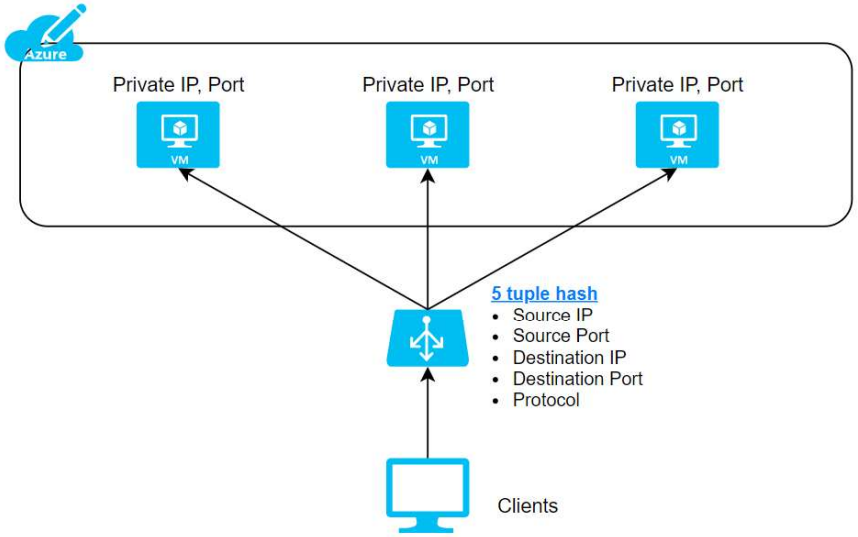
- We specify health probe
- We can also specify session persistence. If this option is enabled, the traffic will be routed to the same VM.



Azure Load Balancer can also be configured to use as follows to map traffic to the available servers:

- **2 tuple (Source IP, Destination IP)**
- **3 tuple (Source IP, Destination IP, Protocol)**
- **5 tuple (Source IP, Source Port, Destination IP, Destination Port, Protocol)**

Please see how the traffic is routed based on the 5 tuples.



Azure Application Gateway

One of the main benefits of the Cloud is elasticity on-demand.

In a traditional datacenter, if there is a peak load requirement of 100 cores from 10-11 am when users login, the machines will always need to have the capacity of 100 cores.

However, in the cloud environment, we will have a single VM with 50 cores at all times and add another VM with 50 cores between 10-11 AM alone. This has reduced consumption by almost 50%.

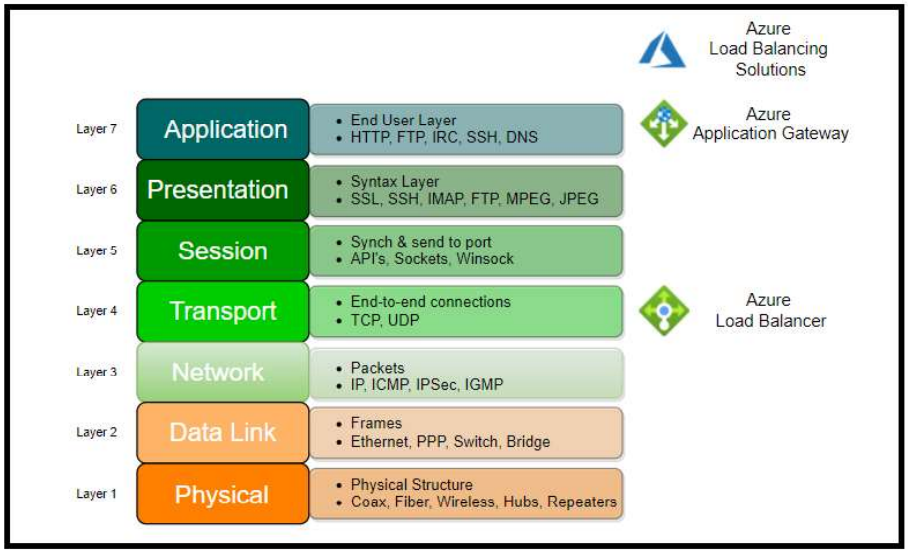
But how do we now distribute the load between the two VMs?

The solution is Load Balancing.

Load balancing can be done at 2 layers in the OSI model. One is at Layer 4 where we will use the Azure load balancer. Here a combination of source and target ip and TCP/UDP Protocol will be used to achieve routing.

The other routing type is at Layer 7, which is the Azure Application gateway. Here the application gateway uses a front-end IP address which is resolved from FQDN via DNS. It has an optional WAF (Web application firewall).

OSI LAYER and the load balancing options within Azure

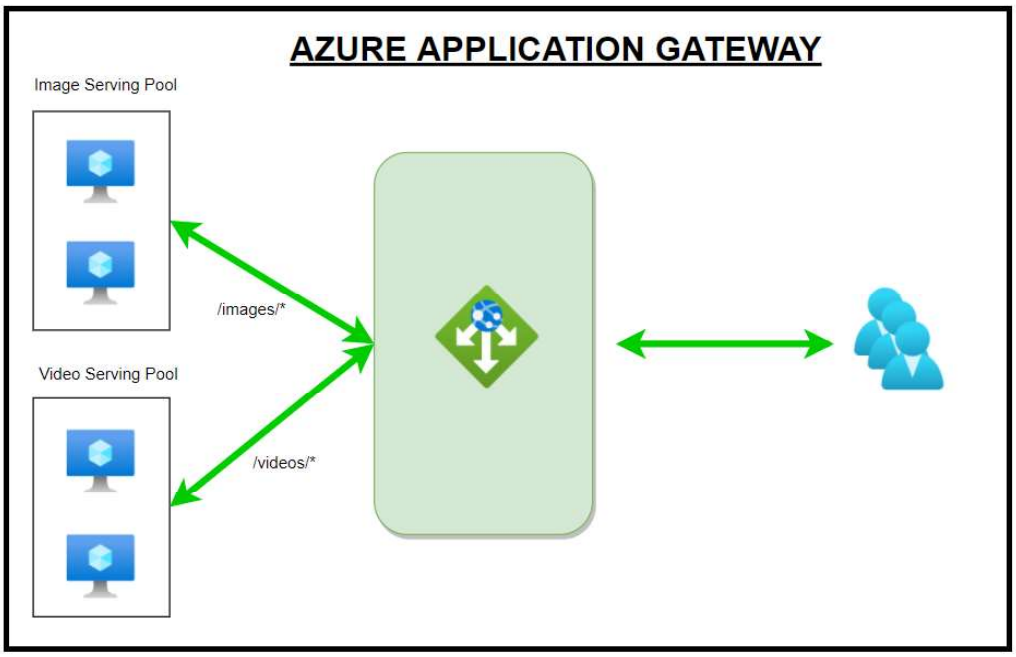


How does the Application gateway work?

Step 1: User sends a request to a website with **FQDN (fully qualified domain name)** – for example, https://demystify.com/videos. The query will be sent to a DNS server, and it will return the IP address.

Step 2: The application gateway will be configured with a listener, a logical entity checking for connection requests. The listener is configured with a front-end IP address, protocol, and port number for connection requests.

Step 3: The application gateway also has a backend Pool/s. The backend pools could be VMs or **VMSS (VM Scale Sets)** or external servers, or Azure App servers. Based on routing rules set up, the traffic will be routed to the appropriate backend servers.



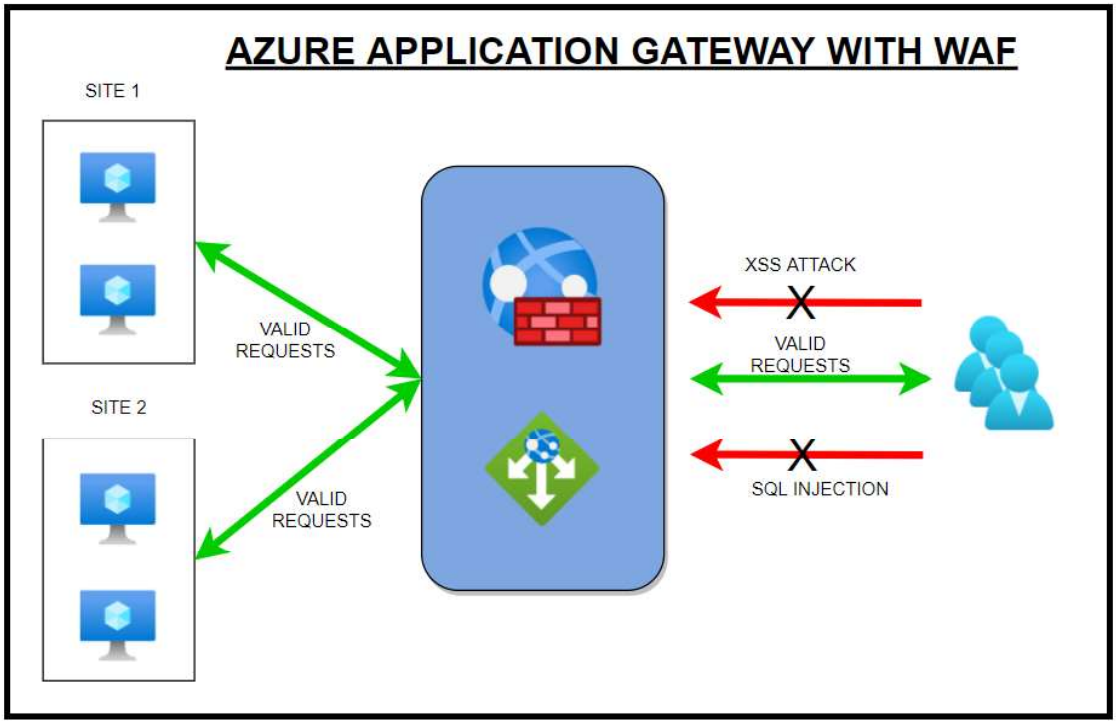
In the above example, you can see the routing rules being processed with url based routing. So when the users type the url https://demystify.com/videos, the gateway sees the videos in the url and sends the traffic to the Video Serving Pool.

Application Gateway with WAF

There is an optional feature WAF that can be additionally added to the application gateway. WAF is based on **Core Rule Set (CRS).**

We need to set up a WAF Policy that has rules. There are two types of rules. One is Managed rule sets which Azure preconfigures. The other is custom rules. Some of the features of WAF are

- Some of the features that WAF provides are preventing SQL injection**/ XSS/ http** protocol violations.
- It also protects against crawlers and scanners. We also can allow or block traffic coming in from certain countries/regions in preview, and it is called **Geo-filter traffic.**
- WAF can be set up in two modes which are Detection or Prevention.
- When WAF is added, the traffic will be evaluated before Step 3 above against the WAF rules.
- If violating traffic is found in Detection mode, the warning will be issued, and traffic continues to flow. In Prevention mode, the traffic will be blocked.


AZURE APPLICATION GATEWAY WITH WAF
SITE 1
SITE 2
VALID
REQUESTS
VALID
REQUESTS
XSS ATTACK
VALID
REQUESTS
SQL INJECTION

Azure Traffic Manager

Azure provides the following services for Delivery.

- *CDN*
- *Front Door*
- *Traffic Manager*
- *Application Gateway*
- *Load Balancer*

While Load Balancers and Application Gateways operate at **Layer 4 and 7,** Traffic Manager operates at a DNS level.

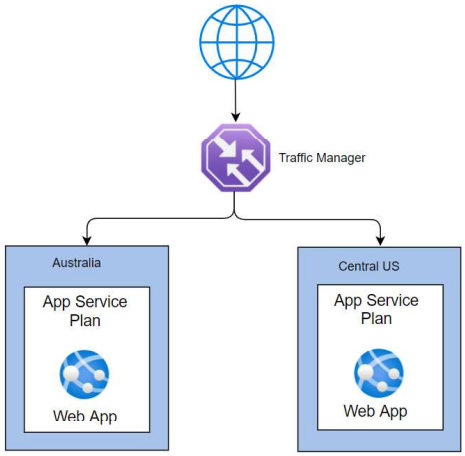
This service will distribute traffic to *public-facing azure services at a global level.* The public endpoints provided are having high availability and quick response.

We can use Traffic Manager to route traffic to regional application gateways at a global level, which could have a load balancer setup for multiple VMs at a database tier utilizing all the services.

Here are some more scenarios:
Application Gateway - to load balance between your servers in a region at the application layer.

Front Door - optimize the global routing of your web traffic and optimize top-tier end-user performance and reliability through quick global failover.

Load Balancer - Network Layer Load Balancing



How does a Traffic Manager work?

The Traffic Manager uses DNS for resolution. It uses this to find the name server. Then it locates the endpoints (which are not disabled) and routes the traffic based on the routing methods specified.

Routing Methods:

Here are the routing methods which we can configure:

Routing Method	Scenario
Priority	*When we have several endpoints, and we want to use one location preferentially, we can use this method having a primary service endpoint for all traffic.* *We can configure one or several multiple backup endpoints in case the primary is unavailable.*
Weighted	*When we want to split and route traffic to different locations, we should use this method. We have to set weights to accomplish this.* *Let's say we want to route traffic equally, we set the weight of 1 and 1 to both the endpoints* *If we give weights of 1 and 2, then the ratio will be 33:66 and one-third of traffic will go to the first endpoint, and two-thirds of traffic will go to the second endpoint.*
Performance	*Let's say that we have 3 locations like Las Vegas, Houston, and Jersey City on 3 sides of the country. We would like end-users to use the "closest" endpoint for the lowest network latency.* *For example, users in New York should connect to Jersey City, which is the closest location. Then we should select this routing method for the lowest latency by choosing the closest endpoint.*
Geographic	*Let's say that there is a requirement that data from a country (Saudi Arabia) has a mandate that data should not cross borders with sovereignty laws.* *We can use this method to direct users to specific endpoints based on where their DNS queries originate from geographically.* *So if a user from this country tried to access it, he would be routed to the servers in his country only.*
Multivalue	*If there multiple servers and we wanted to select multiple servers to select any of the available servers, we can select MultiValue.* *When a query is received for this profile, all healthy endpoints are returned. We can limit the servers returned by setting a max value.*
Subnet	*Use this method to map sets of end-user IP address ranges to a specific endpoint. When a request is received, the endpoint returned will be mapped for that request's source IP address.*

FAQs

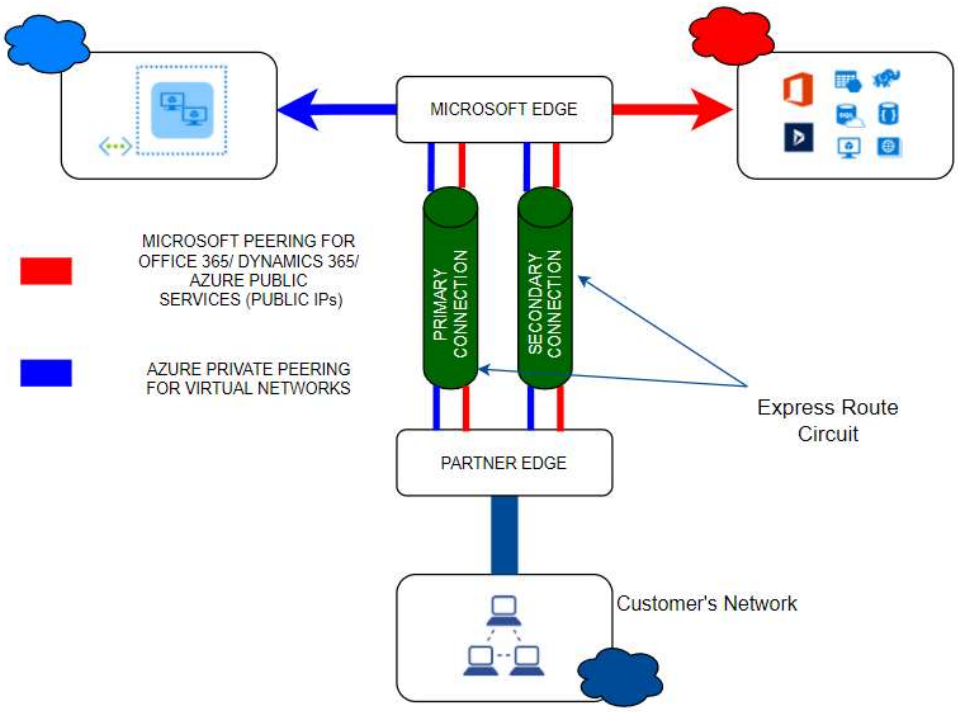
1. **What is the name of the website that will be created when we configure Traffic Manager?**
   Azure will always use azurewebsites.net as a suffix. We cannot change it

2. So how do we use our website like demystify.com?
   You need to create an alias in your DNS zone and point to the Traffic Manager.

Azure Express Route

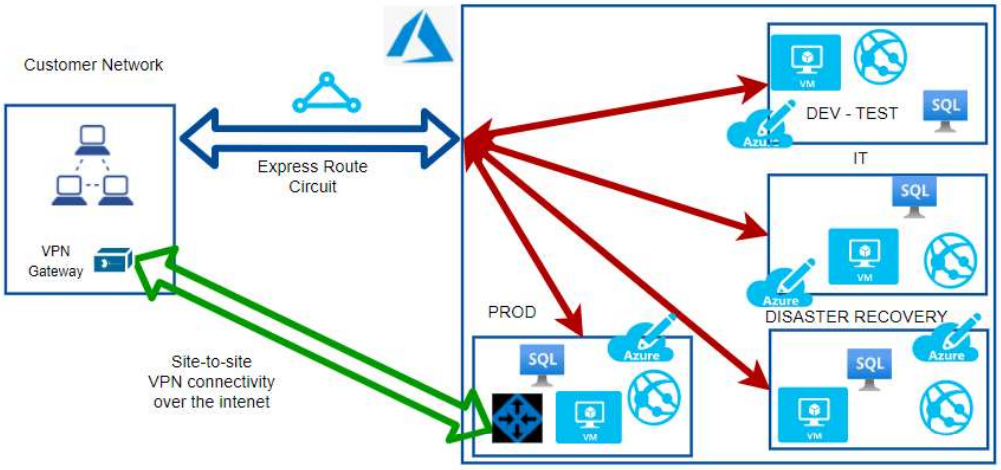
Connectivity to Azure

- There are several ways to connect to Azure. Broadly classifying them, we could either use the internet or have a direct connection.
- While connecting via the internet, we need to use a VPN to connect our infrastructure on premises with the cloud using a **VPN** gateway which encrypts our traffic by creating a tunnel.
- We could choose either a client-to-site VPN which is only one client system connected to the **cloud or site-to-site** VPN where we connect two sites.
- This setup depends on the public internet and we must secure and could have reliability issues.
- Hence it is better to use a dedicated connection between our infrastructure and the cloud with an **Express Route connectivity.**
- We need to locate a connectivity provider. There are several choices available based on location.
- For example, in India, we have *BSNL/AIRTEL/SIFY and in the USA, we have AT&T/ SPRINT/ VERIZON and many more.*



- Express Route connectivity allows us to connect to 2 Microsoft cloud services – **Microsoft Azure Services as well as Microsoft 365 services.**
- Also, we can see from the above diagram that there is an active-active redundant pair of cross connections setup for high availability. We can add further redundancy by adding up to 16 Express route connections.

- Express route has the following bandwidths to choose from based on our requirements:
- **50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, 10 Gbps**
- If we have multiple subscriptions, we can connect all of them to a single Express route connection.
- You can have upto 10 Vnet connections on a standard Express route connection and upto 100 for Premium connection. However please note that all connections will share the same bandwidth.



We could even have a site-to-site VPN for adding redundancy. If there were issues with the Express route, we can failover to the S-2-S VPN.

FAQs

1) **If I have a 100 Mbps circuit, what is ingress and egress capacity?**
   You will have an incoming capacity of 100 Mbps and outgoing capacity of 100 Mbps.
   What is the routing protocol?
   Express route uses BGP (Border gateway protocol)
2) **What happens if there is any maintenance?**
   There won't be any impact. Express route uses an active-active setup and only the circuit will be maintained at a given time.
3) **So where does the connection land on the Azure cloud?**
   We connect to one of the Vnets in a subscription. We can connect upto 10 Vnets in each of the 10 subscriptions max. We need to go for Premium if we would like to add more.
4) **How do we plan for Disaster recovery?**
   Microsoft recommends 2 Express connectivity to avoid a single point of failure. We could also set up a Site-to-site VPN instead of a second circuit.

Azure VPN Gateway

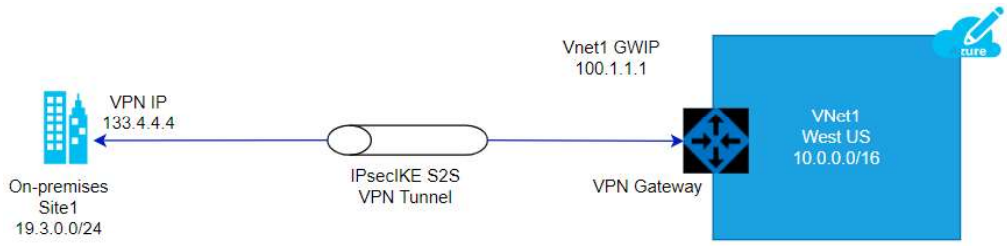
This is one of the methods that allows inter-site connectivity. Express route is the preferred connection as it has higher bandwidth and is in active-active mode. Smaller enterprises can choose VPN gateways or one could use VPN Gateway as a backup to Express route connectivity.

The VPN gateways are set up over the Public internet. Hence the traffic needs to be encrypted. **IPSEC** is used as the tunnelling protocol which creates a secure tunnel through which the data travels. Even if the traffic is intercepted, it cannot be decrypted.

VPN Gateways types

- **Site-to-Site VPN Gateways**
  - Here the On-premises will be a site and Azure VPN will be another site. We can connect multiple VNets.
- **Point to Site VPN gateways**
  - Here we connect a single client machine from on-premises to the Azure VNet.
  - We can use the same connection on multiple clients by exporting the configuration from the existing client.
- **Internal Gateway between Azure networks**
  - This is a special use case where we want to encrypt traffic between Azure Vnets.

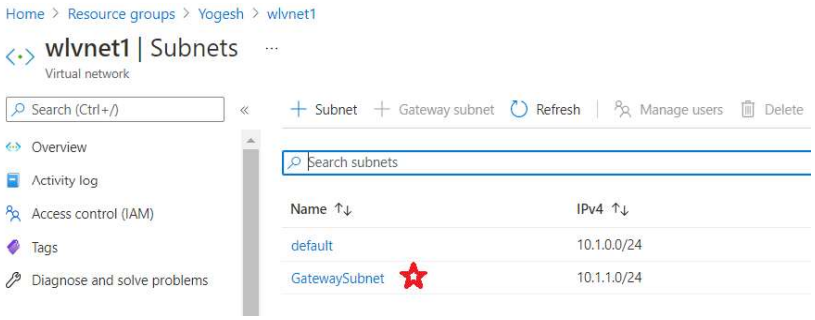
VPN Gateway Architecture



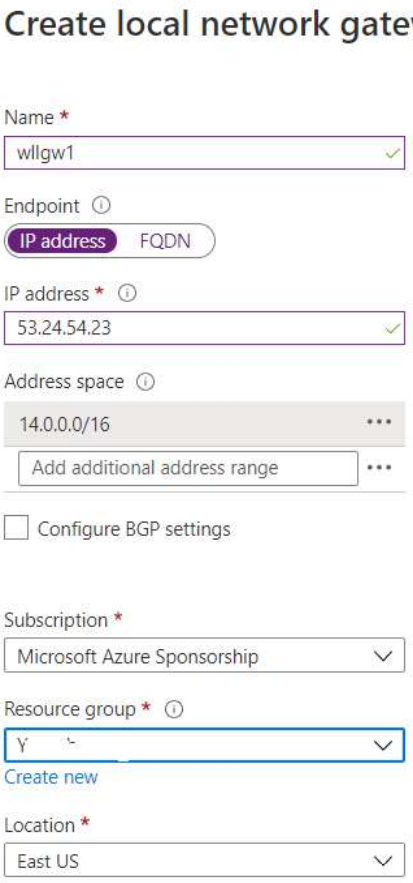
Steps to establish VPN Gateway

1. **GatewaySubnet**
   a. We need to create a subnet with the name "**gatewaysubnet**" for the setup
   b. If we are creating a Vnet, this subnet gets created automatically.

Please see the diagram below which shows the gateway subnet. This was created implicitly when the wlvnet1 was created as part of the Vnet gateway creation.



2. **Local Network Gateway**
    a. We need to obtain a Public IP address from the on-premises admin team and use that as the endpoint.
    b. See the IP address given as **53.24.54.23**. This is the ip address of the router on-premises

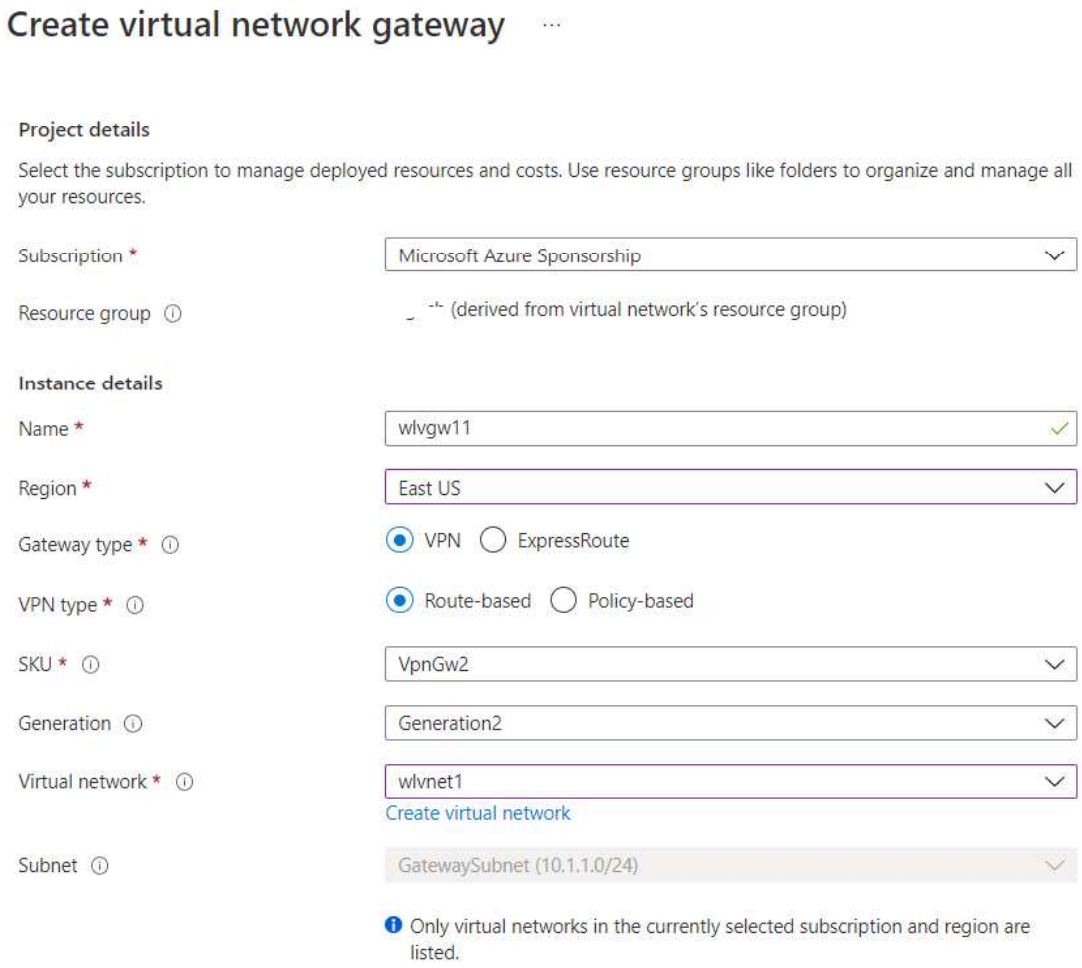


3. **Virtual Network Gateway**
    a. We need to create the virtual network gateway with the following inputs
        i. **Gateway type** – in our case, we are going to use VPN
        ii. **Vpn type** – could be either Route-based or Policy-based. Please note that we cannot change the type once it is created.

We need to delete the gateway and recreate it to make the change. Policy based is the most common type

iii. **SKU** – There are several SKUs. Please note that Basic is considered legacy and not recommended.

iv. **Subnet** – As mentioned, the name should be GatewaySubnet.



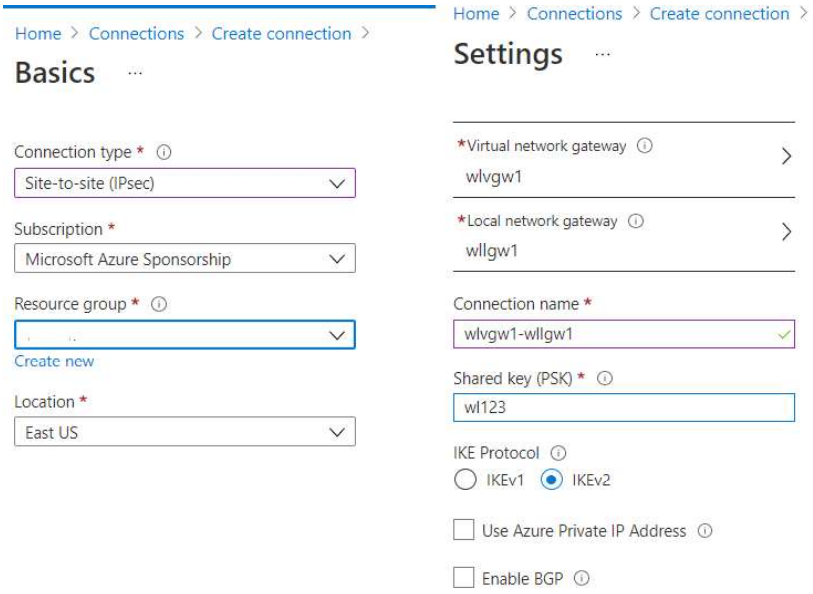
4. Connection

a. Once the prerequisites are fulfilled with the creation of Gateway Subnet, Local Network Gateway and the Virtual Network Gateway, we can create connection as follows

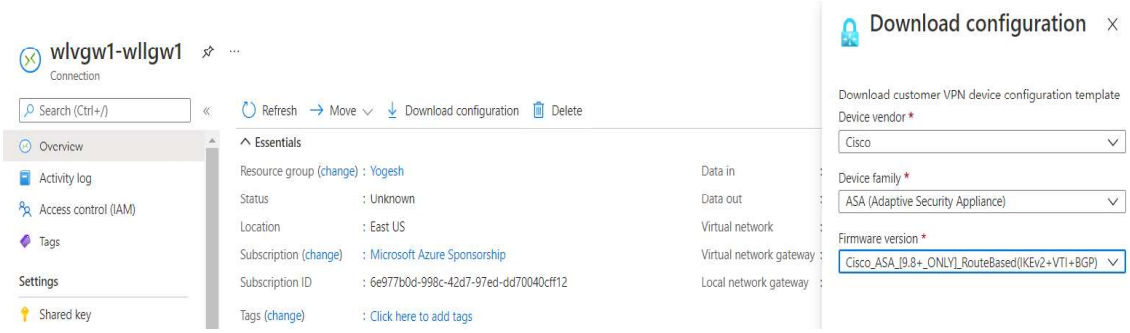
i. **Connection type** – the options are Vnet-to-Vnet, Express Route or Site-to-Site. In our case, we choose Site-to-site which uses the IPsec tunnelling protocol by default.

ii. **Bidirectional Connectivity** – Connections are usually unidirectional. We can select bidirectional to choose 2-way communication

iii. **Shared Key(PSK)** - We need to create a password here and need to share this with the on-premises admin to configure from their side.

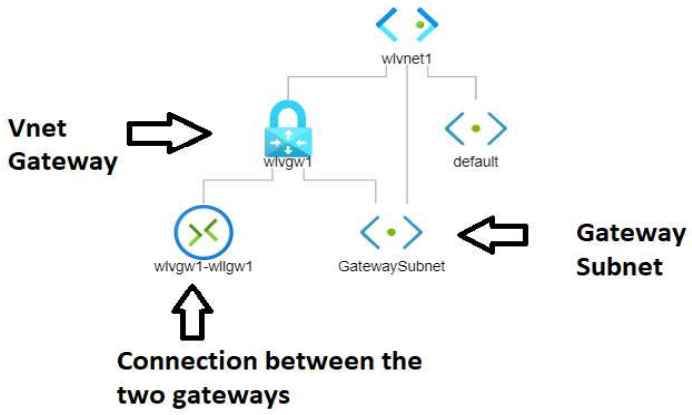


5. **On-premises setup**
    a. Once the setup is complete, we can download the configuration to be shared to the on-premises admin.
    b. We need to get the router model and select the same from the dropdown list and download the configuration and share with the admin along with the shared key.



Topology

We can check for the topology from the network watcher – topology blade.



Azure CDN

What is a CDN?

CDN stands for **Content Delivery network**. It is an architecture of distributed network of servers that can efficiently deliver web content to users.

CDNs will cache the content on edge servers in the POP (point of presence) locations keeping the content closer to the users thereby minimizing latency. This is made possible by using the existing network infrastructure of the CDN provider.

Let's say that a company demystify has a headquarters in NY, USA and branches in CA, USA and Bangalore, India.

The Servers are located in NY and we have a user logging in from Bangalore, India. The data needs to traverse the network and this will cause latency.

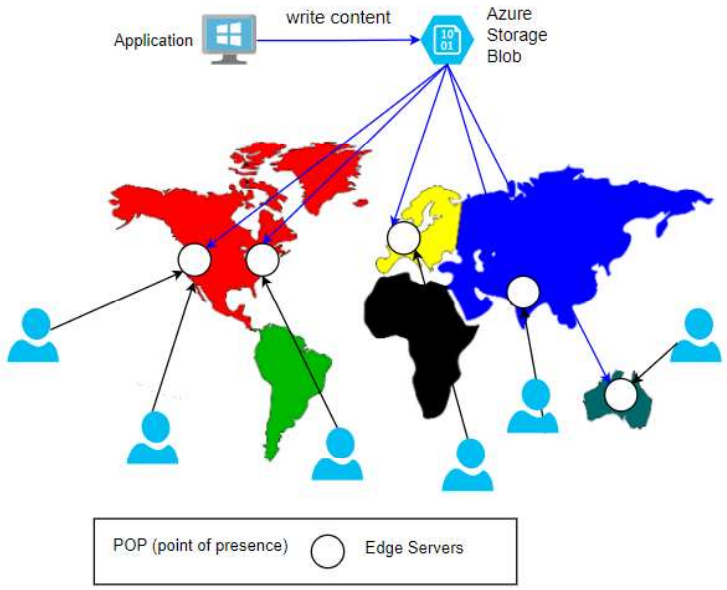
The solution here is to use CDN and use the Bangalore location to cache the data. Now, the user will be able to retrieve the data from Bangalore.

Please note that the data is not stored permanently on the edge location. This is to ensure that data does not go stale and it is current.

So we may set a cache interval of 24 hours and every day, the data will be retrieved from the Origin Server (NY, USA) and cached on the edge locations.

Also, if the data is not available (first time accessing) or if the cache has been marked as invalid, the data will be fetched from Server and sent to the user and cached on the edge location.

The next time, the request will be fulfilled by the edge server. This also reduces the load on the Origin server.



FAQs:

1) **How long is the data cached on the edge Server?**
   The TTL (time to live) by default is 7 days. This can be configured as per the application requirements. Once TTL expires, the cache will be marked as invalid.

2) **What type of azure servers can serve as Origin Servers to get source data?**
   Azure Web App, Azure Cloud Service, Azure Storage account, or any public web server.

3) **What are the CDN products available?**
   Azure has its own product. Besides that, it has tied up with Akamai and Verizon. Here are the offerings:

   a. **Azure CDN Standard from Microsoft**
   b. **Azure CDN Standard from Akamai**
   c. **Azure CDN Standard from Verizon**
   d. **Azure CDN Premium from Verizon.**

Please note that not all products might be available at all locations. You will need to check the product availability for your location.

4) **What are some of the additional features?**
   a. **Dynamic site acceleration(DSA)**
   b. **Video streaming optimization**
   c. **Customizable, rules based content delivery engine**
   d. **HTTPS support with CDN endpoint**
   e. **Compression encodings**

5) **Who are the market leaders for CDN? - PFB**

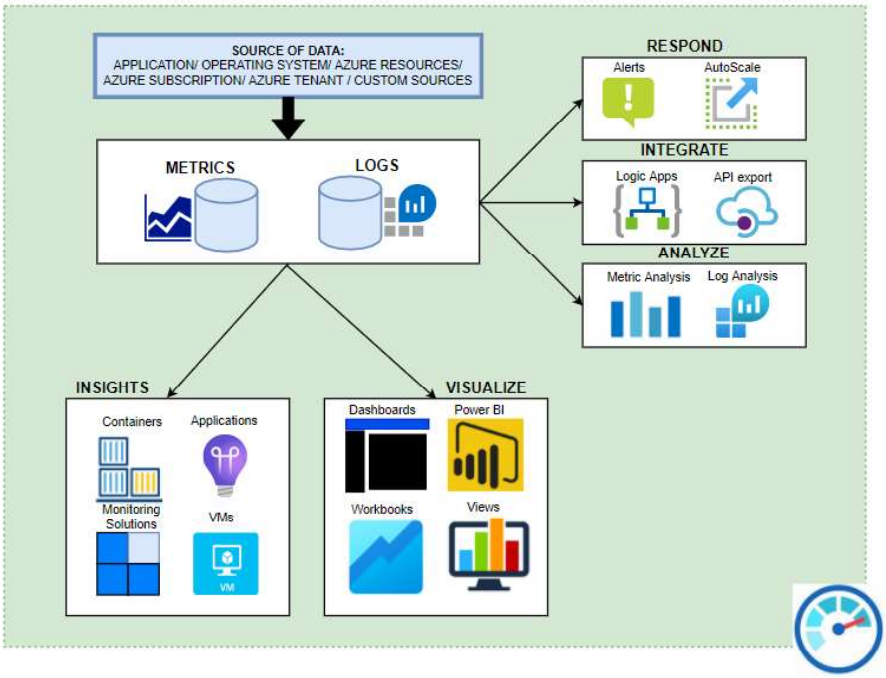
Top Competitors	Market Share	# Websites
jQuery CDN	38.60%	19,13,841.00
CloudFront	24.57%	12,18,186.00
BootstrapCDN	8.88%	4,40,178.00
Amazon S	37.79%	3,86,324.00
Vimeo CDN	5.71%	2,82,933.00
CDN JS	4.16%	2,06,402.00
OSS CDN	3.59%	1,78,093.00
CloudFlare	2.56%	1,27,104.00
Microsoft Ajax CDN	1.97%	97,471.00
Akamai	1.67%	82,949.00
MaxCDN	0.49%	24,381.00

Azure Monitor

- Azure Monitor is a free service that helps increase performance and availability. We could collect telemetry data from Azure as well as on-premises.
- We could collect the metrics and logs from our resources like VMs. We could even collect more detailed logs by enabling guest diagnostics and collect OS level information.
- We can also integrate additionally with **SIEM and ITSM** tools. We could also send data via event hubs or other services.
- Metrics are available at each resource level or they can be collectively seen at the Azure Monitor. This way Monitor acts as a central location for all our monitoring needs like Metrics, logs, alerts and activity logs.
- We also have a section on Insights where we can see more intelligent information for various resources like *Applications, VMs, Storage Accounts, Containers, Networks, SQL (Preview), CosmosDB, KeyVault, Azure Cache for Redis.*
- We could also see a map of our application and understand how the different components work together.

At a high level, we do the following:

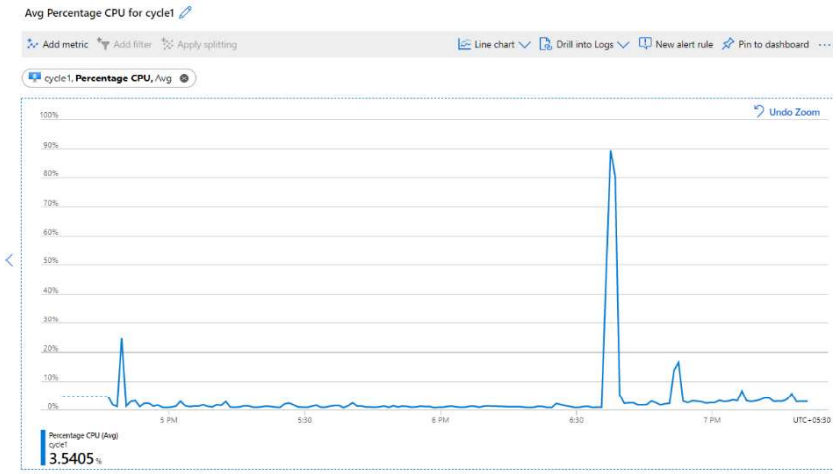
1. **Monitor & Visualize Metrics**
2. **Query & Analyze Logs**
3. **Setup Alert & Actions**



Here are some of the components which make up the Azure Monitor

1) **Inputs** –
   a. **Logs** – these are the logs generated by various resources like VMs/ Databases etc.,

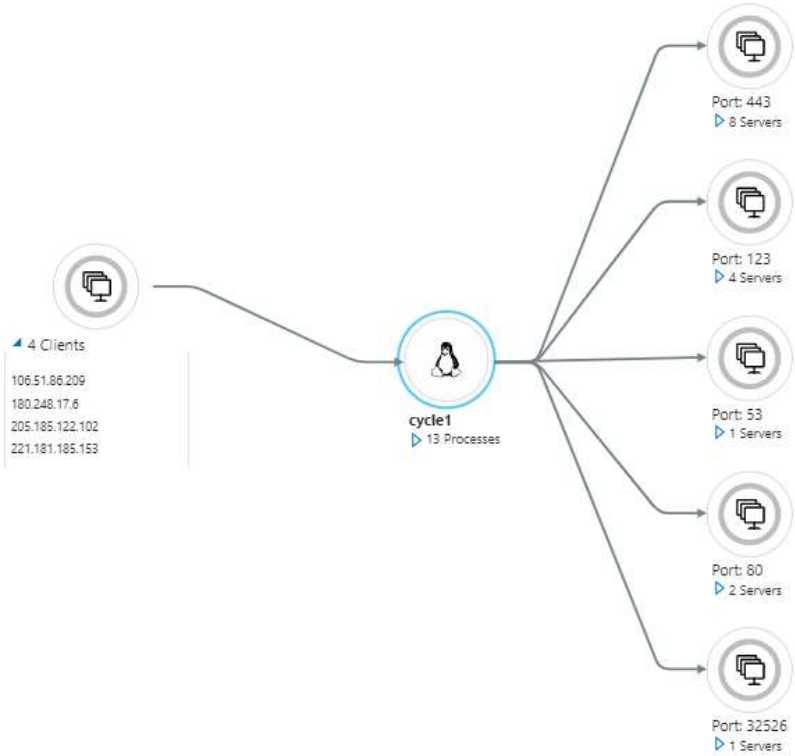
b. **Metrics** – Metrics provides numbers like *CPU percentage, Network data in/out* which helps us understand performance.
The metrics are stored in a time series DB which helps understand real time scenarios. With metrics, we can set triggers to scale the resources up and down. Please see a metric chart below on CPU percentage usage:



2) Insights

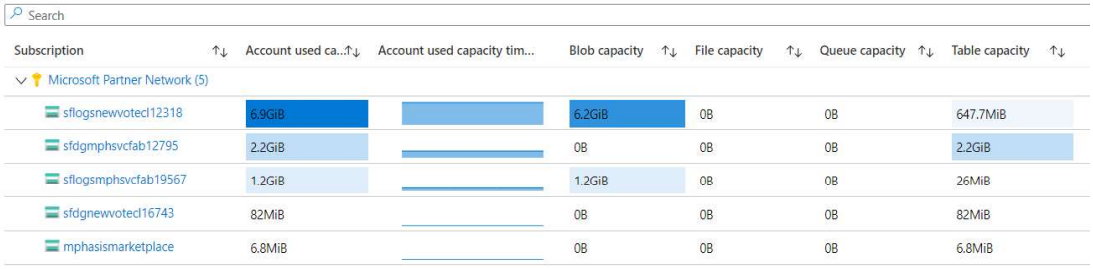
a. With Insights, we can get a deeper view into the resources. We could see a map of the resources and get an overall view. Please see below some insights:

APPLICATION MAP:

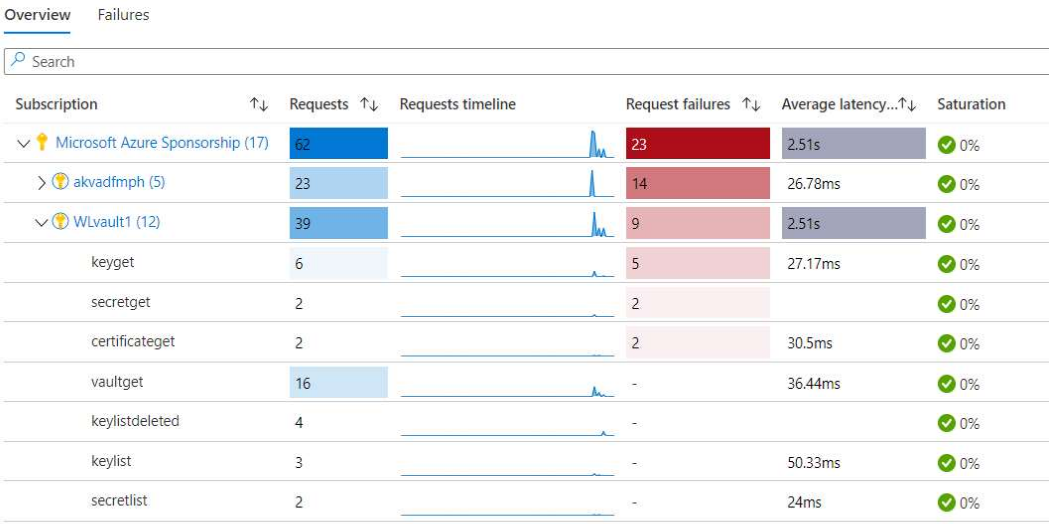


STORAGE OVERVIEW

Overview Capacity



KEYVAULT INSIGHTS



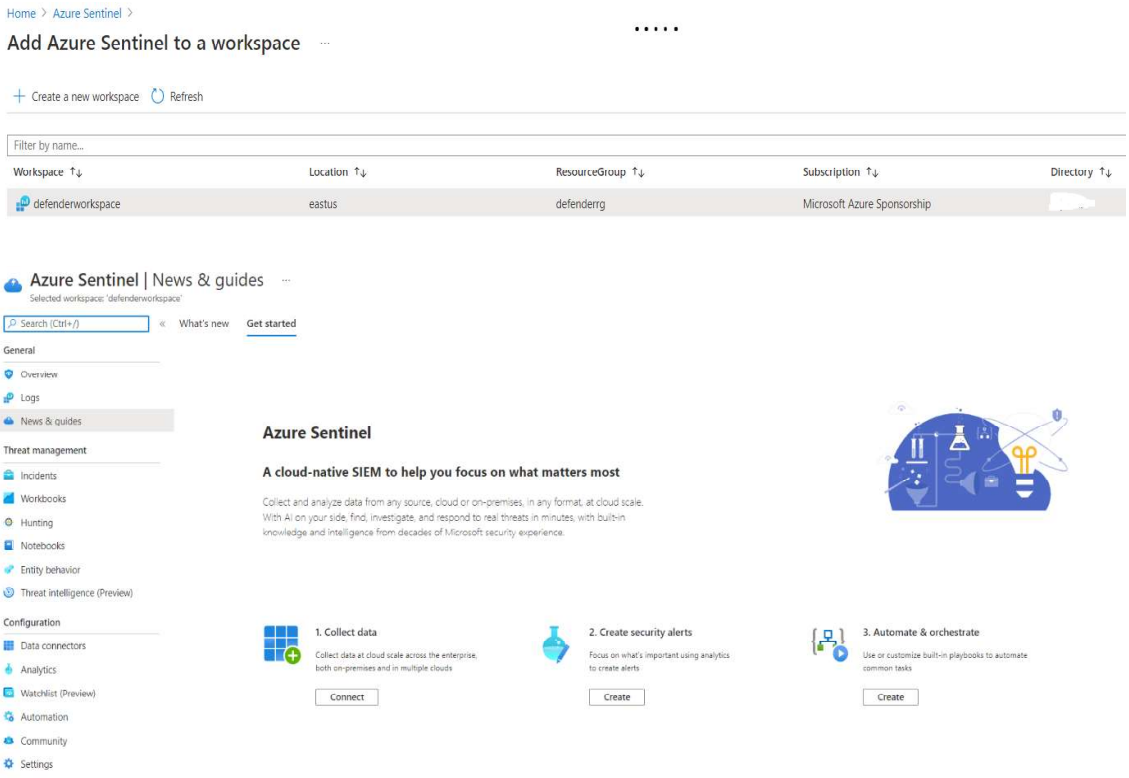
3) **Analyze**
    a. **Log Analytics** – We can work with log data from multiple sources with log analytics. We can perform complex queries with *KQL (Kusto Query Language)*. We can analyse and act on that data.
    b. **Metric Analysis**
4) **Visualize**
    a. **Metrics explorer** – interactively work with metric data with metric explorer
    b. **Workbooks** – We can use a combination of text, metrics, log queries and parameters into interactive reports. There are several built-in workbooks available for use.
    c. **Dashboards** – We can add metric graphs and queries output and create dashboards
5) **Respond**
    a. **Alerts** - When there is any issue, then we will get alerts proactively and we can automatically run *functions, runbooks, webhooks or logic apps.*
    b. **AutoScale** – With the metric as inputs, we can set up the system to scale up or down automatically.

Azure Sentinel

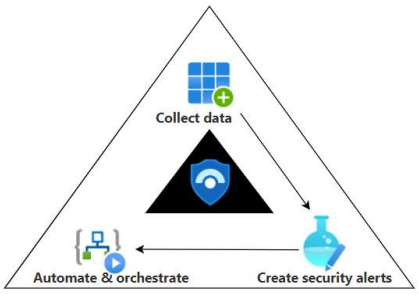
Azure Security Centre provides us with basic visibility and Analytics but Azure Sentinel goes beyond this and provides complete cybersecurity whereby it is able to provide **visibility/ analytics and Hunting/ Incidents** and finally responding to the incidents with automation.

Initiating Sentinel

- To initiate the Sentinel service, we need to create or connect **Log Analytic Workspaces**. Once done, you will see the Sentinel Panel.

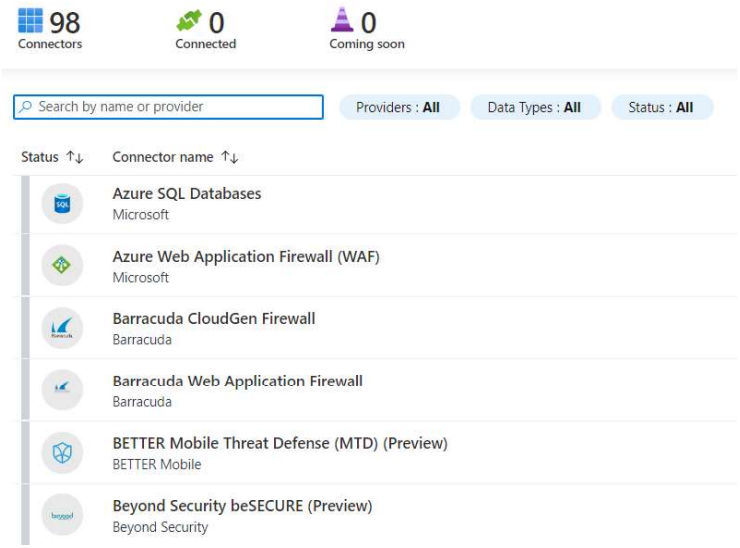


Here are the phases of Azure Sentinel:



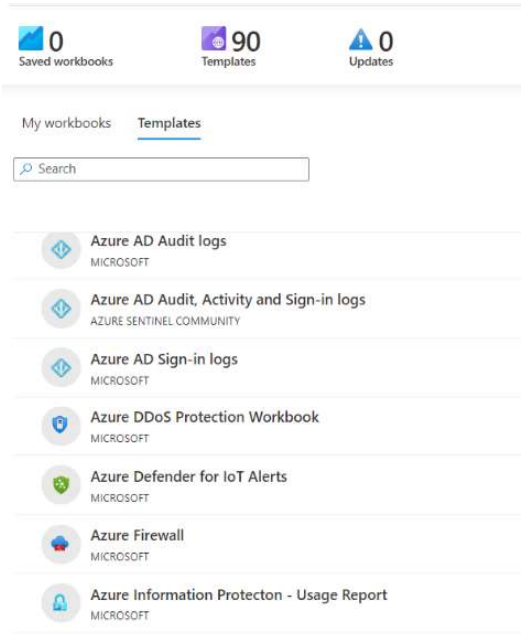
1. **Collect data**
   a. Azure Sentinel can collect data at cloud scale across the enterprise, both on-premises and in multiple clouds.

b. There are **98** data connectors available as of today like **Azure/ AWS** etc and we can connect to these sources and receive the data. **Examples** of Azure data are *azure sign-in activity from Azure AD.*

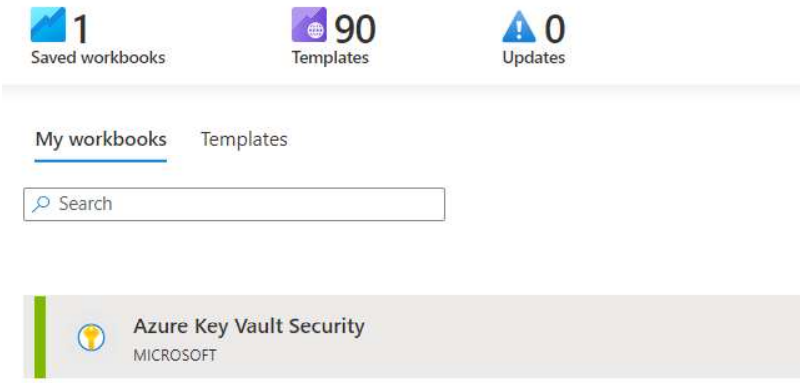


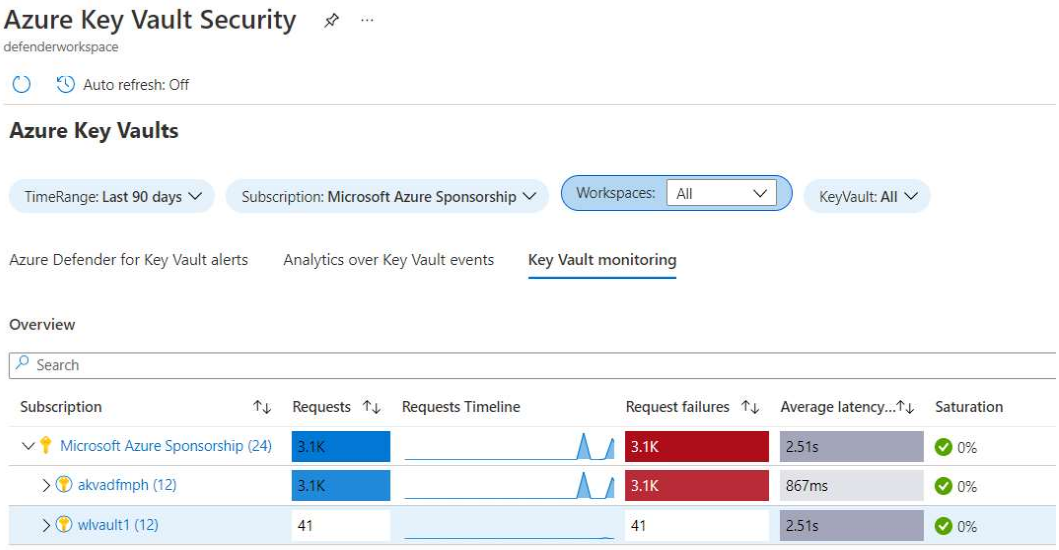
2. **Create Alerts**
   a. Once the data is collected, we can run queries against the data. We focus on what is important using the analytics and create suitable alerts.
   b. There are prebuilt workbooks available which can be selected and used to get insights.

Workbooks selection



- We need to select the template and save it and we will be able to get the details.





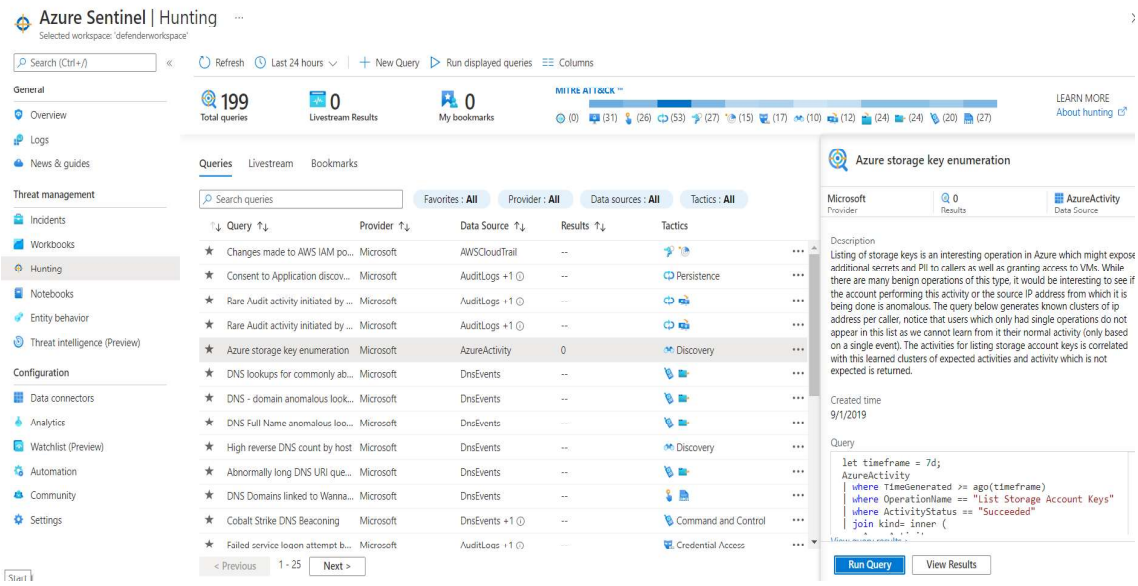
3. Automate and Orchestrate

- **a.** We build automation rules which will automate incident configuration. We could trigger playbooks to handle security alerts.
- **b.** We create rules which will trigger the playbooks to be run automatically based on the conditions

Hunting Feature

Azure Sentinel has the hunting feature where we could go further and search for various activities like listing of storage keys or high **DNS queries** etc.,

This will help us identify attacks targeted and we could go and proactively block the malicious activity.



Sentinel Community

There is a sentinel community where we can get different types of resources like *Workbooks, Analytics rules, Hunting queries, and Playbooks.*

Azure Sentinel pricing

Billing is based on the volume of data ingested for analysis. Azure Sentinel offers a flexible and predictable pricing model and we could pay either with **Capacity Reservations or Pay-as-you-Go**.

With Capacity Reservations, we can get as much as **60%** less as compared to Pay-as-you-Go.

Advanced Threat Protection

Azure has a product called Azure defender for SQL which is a unified package for advanced SQL security capabilities. This is designed for the database offerings viz., *Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics*.

Some of the functionality of the tools include

- **Discovering and Classification of sensitive data**
- **Identifying and mitigating potential database vulnerabilities**
- **Detection of anomalous activities that could be a potential threat**

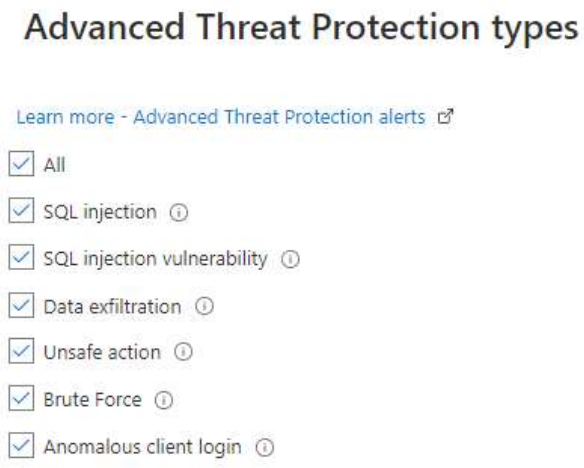
The tool does 2 major activities. One is *Vulnerability assessment and the other is Advanced Threat Protection.*

Under ATP, the following features are available:

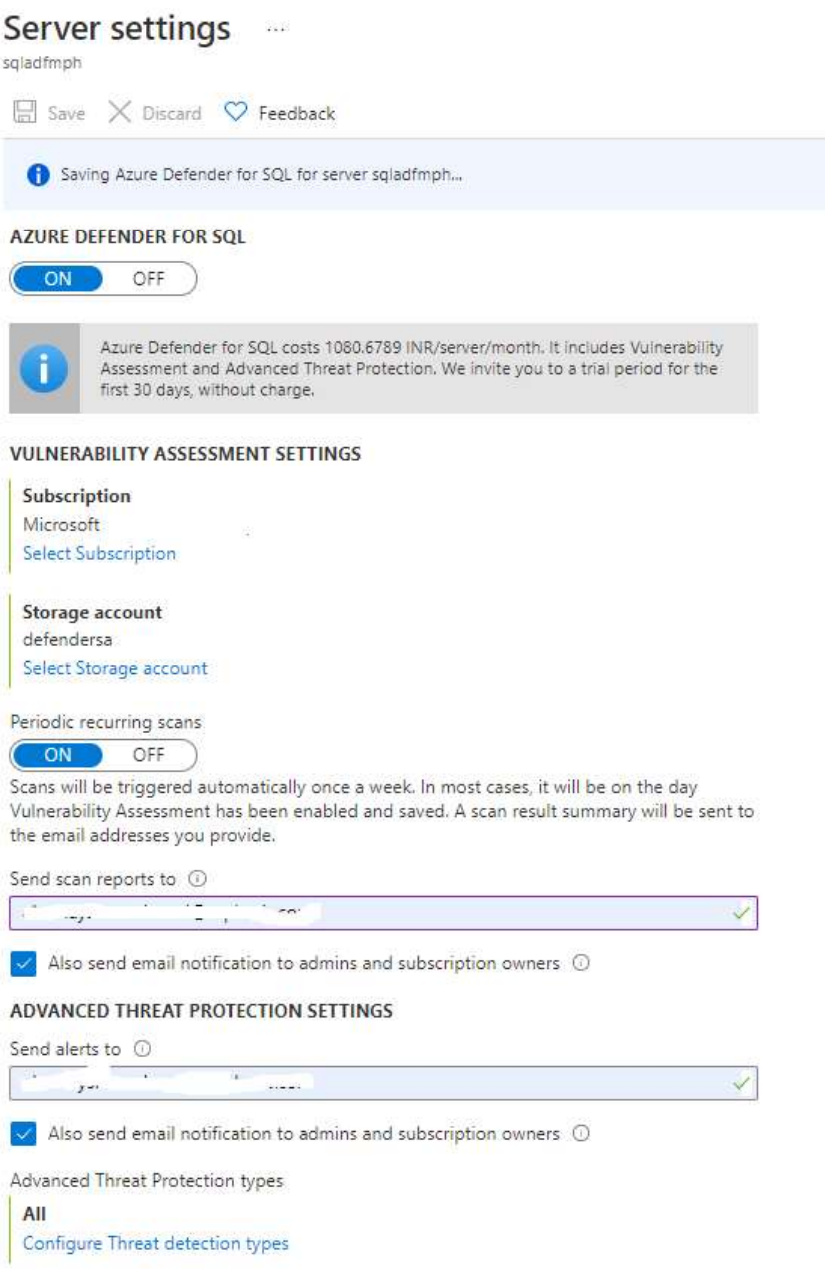
- **Detect anomalous activities**
  - Unusual/potentially harmful attempts to access or exploit your database.
- **Continuous monitoring of database for suspicious activities**
- **Immediate security alerts on**
  - Potential vulnerabilities
  - SQL injection attacks
  - Anomalous database access patterns.
- **Recommend action on how to investigate and mitigate the threat.**

How to Enable ATP:

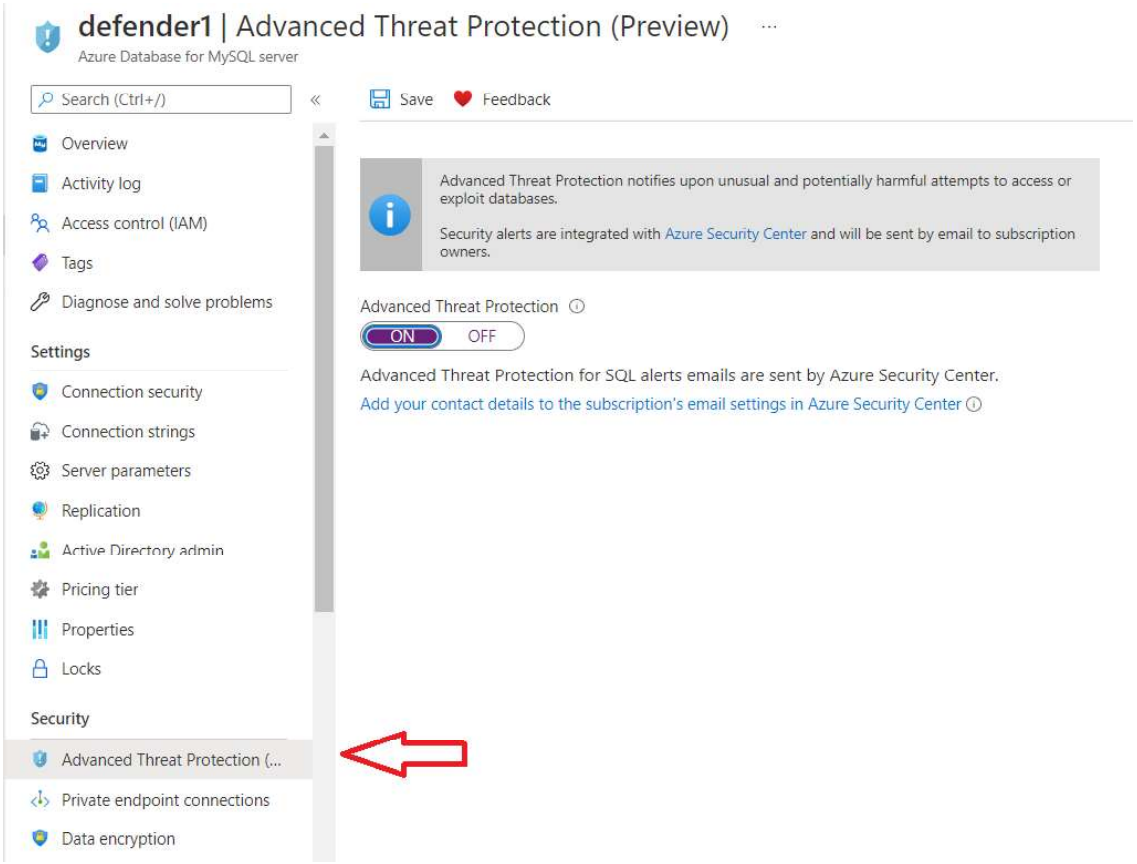
- We can enable/disable different type of alerts under ATP



Under Security settings of the SQL server, we can enable ATP.



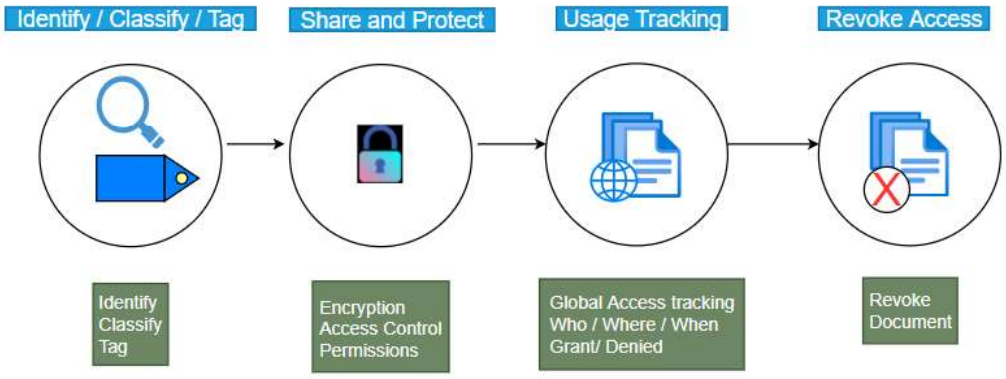
- We can see the ATP alerts in the Security Center as it is integrated with it.
- The MYSQL Database server can be configured under Security Option.


defender1 | Advanced Threat Protection (Preview)
Azure Database for MySQL server
Search (Ctrl+/)
Save
Feedback
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Connection security
Connection strings
Server parameters
Replication
Active Directory admin
Pricing tier
Properties
Locks
Security
Advanced Threat Protection (...
Private endpoint connections
Data encryption
Advanced Threat Protection notifies upon unusual and potentially harmful attempts to access or exploit databases.
Security alerts are integrated with Azure Security Center and will be sent by email to subscription owners.
Advanced Threat Protection
ON
OFF
Advanced Threat Protection for SQL alerts emails are sent by Azure Security Center.
Add your contact details to the subscription's email settings in Azure Security Center

Azure Information Protection

AIP helps us classify and protect our documents based on the sensitivity of the data. AIP is based on **Azure Rights Management (RMS)** which is a cloud based protection technology.

We can look at the lifecycle of data below:



Lifecycle of data

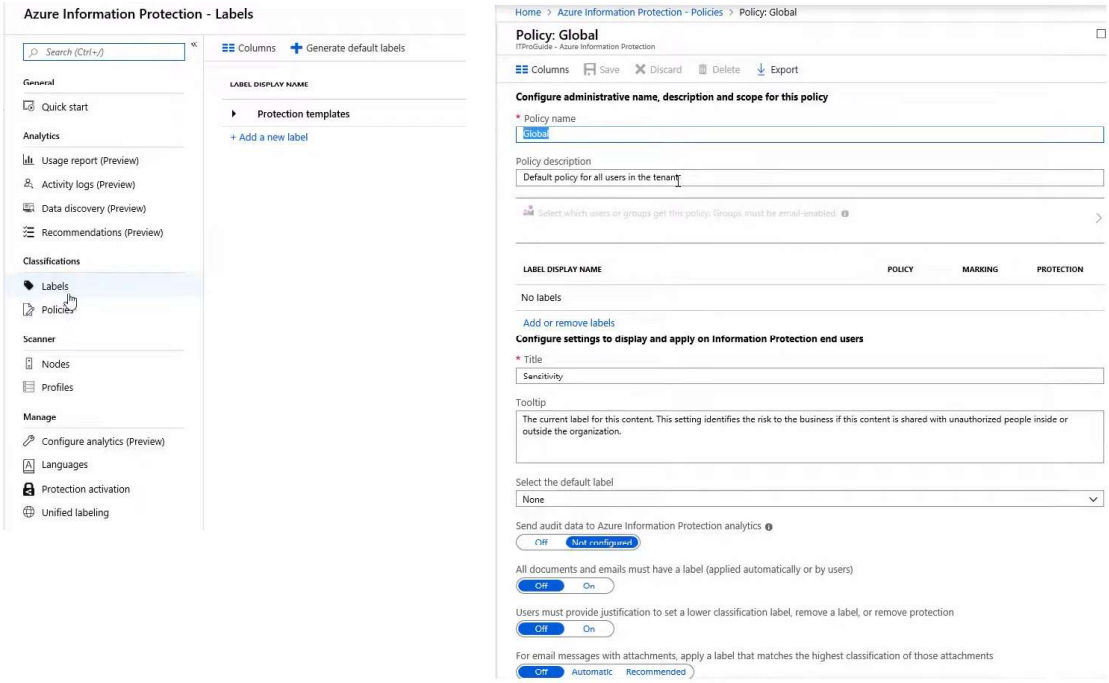
1) **Identify/Classify/Tag**
    a. In this phase, we identify the data that needs to be protected. There are two types of data in an organization. *Structured and unstructured*.
    b. Data that resides in a database can be classified as structured. Data that resides on the servers and user systems can be classified as unstructured. We identify the unstructured data
    c. Then we classify the data. Simple classification could be Public data which is *non-personal and non-confidential.* Likewise we could have confidential data.
    d. Then we need to tag the data

2) **Share and Protect**
    a. Once the data is classified and tagged, we encrypt the data. We could either use cloud key or we could use our own keys
    b. Then we grant or revoke access to the data. We could grant *viewing/ editing* permissions to different groups as needed

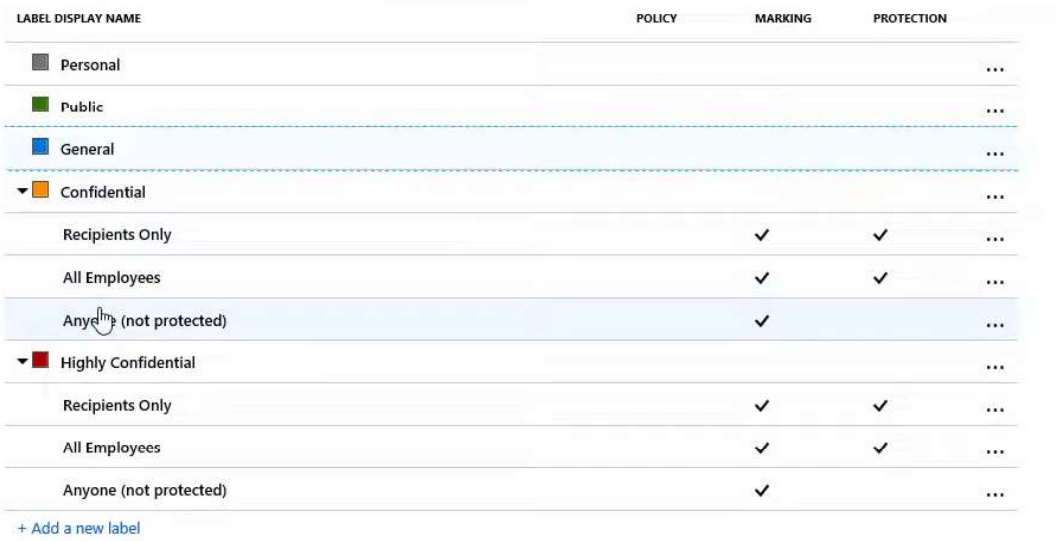
3) **Usage tracking**
    a. With RMS, we could track the usage of data
    b. We can see who accessed the data from which location and when
    c. We can grant or deny access

4) **Revoke access**
    a. Once we revoke the access to the data, the users who had access before cannot access the same data going forward.

- Below diagram shows the Labels and also the Global Policy.



- The Labels are available as default as seen below. We can create custom labels if needed to suit our needs.



AIP lifecycle:

Here are the practical steps to the same

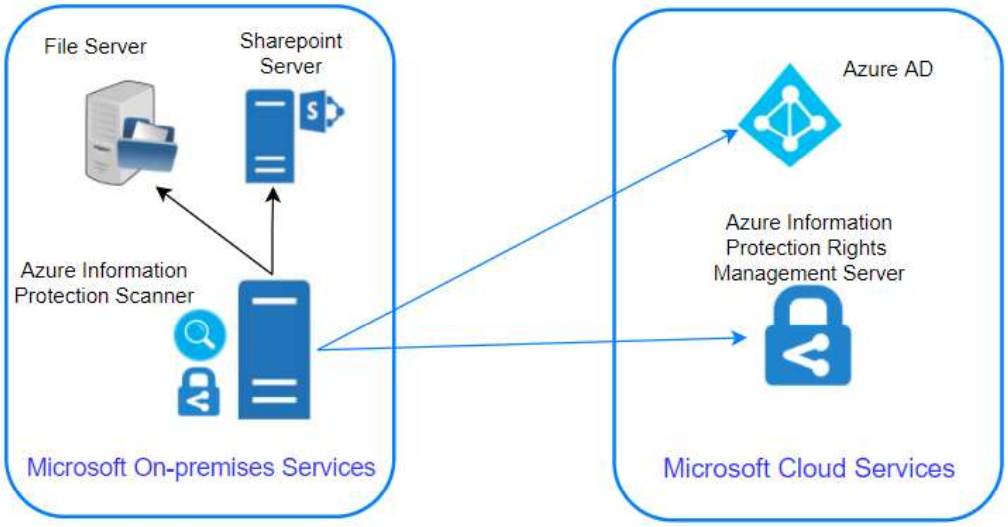
1) **Subscribe AIP** – We need to have suitable licenses like *AD Premium P1/P2*. There are other enterprise licenses also like E5.
2) **Azure AD** – We need to integrate with our AD

3) **AIP Label and Policy** – We need to create labels and then add them to our Policies. There is a global policy available by default. We can further customize or add our own Policy.
4) **Install AIP client** – We need to install AIP client on all servers that we want to manage
5) **Create custom label** – We create custom labels if needed
6) **Revoke Access** – We review and revoke access ending the life of the data managed

AIP Scanner

For On-premises setup, we need to install AIP Scanner which does the following:

1) **Discover**
2) **Protect**
3) **Classify**
4) **On-prem repository** – Any folder/drive like c: etc is considered as a repository



Requirements for AIP Scanner – Windows Server and SQL server

Steps for setting up AIP scanner

1) **Create Profile**
   - Settings like manual / automatic scan
   - Policy enforcement
   - Add Repositories like c:/, f:/docs
2) **Install SQL server**
3) **Run Powershell command and setup AIP Scanner**
   - `Install-AIPScanner -Sqlinstance "instance name" -Profile "profile-name"`
4) **Setup Access token**

Azure DDoS Protection

What are DoS and DDoS?

DoS stands for **denial of service** and **DDoS** stands for **distributed denial of service**.

Scenario:

*Let's say that you have a web server serving web traffic and you are a medium enterprise handling **1000** requests per second. If any malicious entity sends **100,000** requests per second, your server will be busy trying to respond to the 100K requests and unable to serve the regular customers. This is called **Flooding**.*

*Often the load will be so heavy that it will cause the server/machine to crash. This is called **denial of service** where customers are denied service by rendering the server unusable.*

*Imagine the same **100K** requests coming from multiple servers where malicious entities do a coordinated attack with multiple servers. This is called **distributed denial of service** where multiple servers hit a given target to bring it down. We have seen attacks feeding as much as **800 Gbps** which can bring the biggest servers down.*

Azure DDoS

It provides protection against DoS attacks with always-on monitoring and automatic network mitigation.

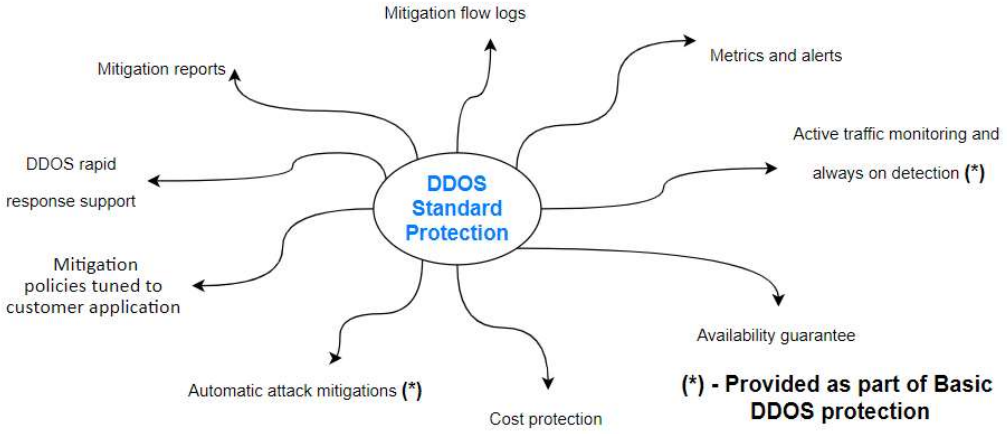
There are two levels of Service – One is **BASIC** and the other is **STANDARD**. Basic plan is free and enabled by default. After all, Azure needs to protects its resources 😊

Basic plan, like the name says, provides only basic services *(always-on monitoring and automatic network mitigation).*

Standard protection provides multiple features. This could cost you as much as **3000$** to protect about 100 resources like

- *Azure firewall, App Gateway/WAF*
- *VMs, AKS*
- *SQL, CosmosDB, Storage, App Services etc.*
- *Vnet*
- So, let's say that a **DoS/DDoS** attack occurs. The Cloud is resilient usually due to the elasticity and if you have good autoscaling, then the cloud resources will keep scaling up like VM spinning up, App Service scaling up etc.
- As a result, you will have a lot of traffic and you must be aware that while ingress traffic is not charged, consumers pay for egress traffic.
- So you will land with a huge compute bill and egress data charges.
- If we had the DDoS standard protection plan, we would be issued credit for the excessive charges if the plan failed to protect us.

Here's the list of services provided.



Some of the features of DDoS Standard protection are:

- **DDoS Rapid response** – We can engage the **DDRT** (DDoS Rapid Response Team) for attack investigation and analysis
- **Cost Guarantee** – As discussed, we will be issued a service credit for the application scale out and excess data transfer
- **Attack alerting/Metrics** - Alerts can be configured to be notified at the start/stop and logging will be done and metrics provided.
- **Extensive Mitigation Scale** – This works at a global scale and is highly scalable and can mitigate over 60 types of attacks.
- **Multi-layered protection** – It can protect at different layers (layer 3/4/7)
- **Adaptive tuning** –Let's say there is unusual traffic from an IP and it is determined as anomalous by the DDoS cognitive services, ddos protection will automatically deny traffic from the IP and block it.

In addition, we can have our own monitoring to alert when a DDoS attack occurs.

We can set up this rule in Azure Monitor to notify us that ddos mitigation has started. We can set up an action group and take actions like notifications, isolating the resource for forensics etc.,

```
AzureDiagnostics
| where Category == "DDoSProtectionNotifications"
| where type_s == "MitigationStarted"
```