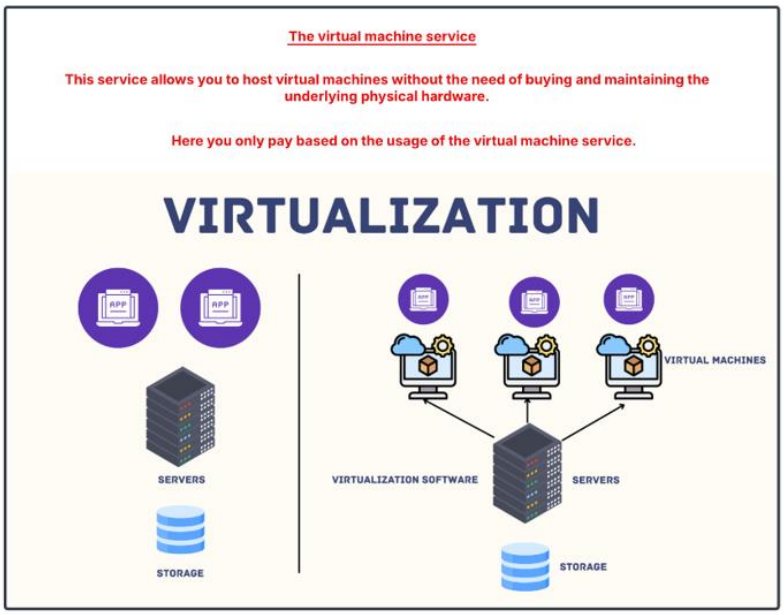
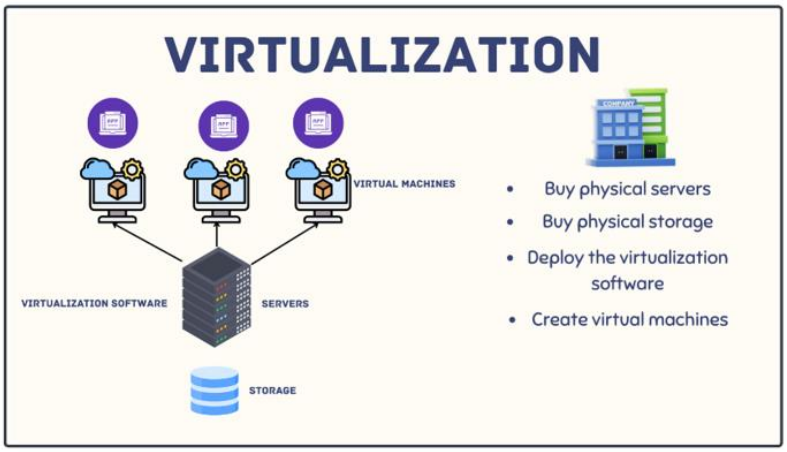
Deploy and Manage Azure compute resources

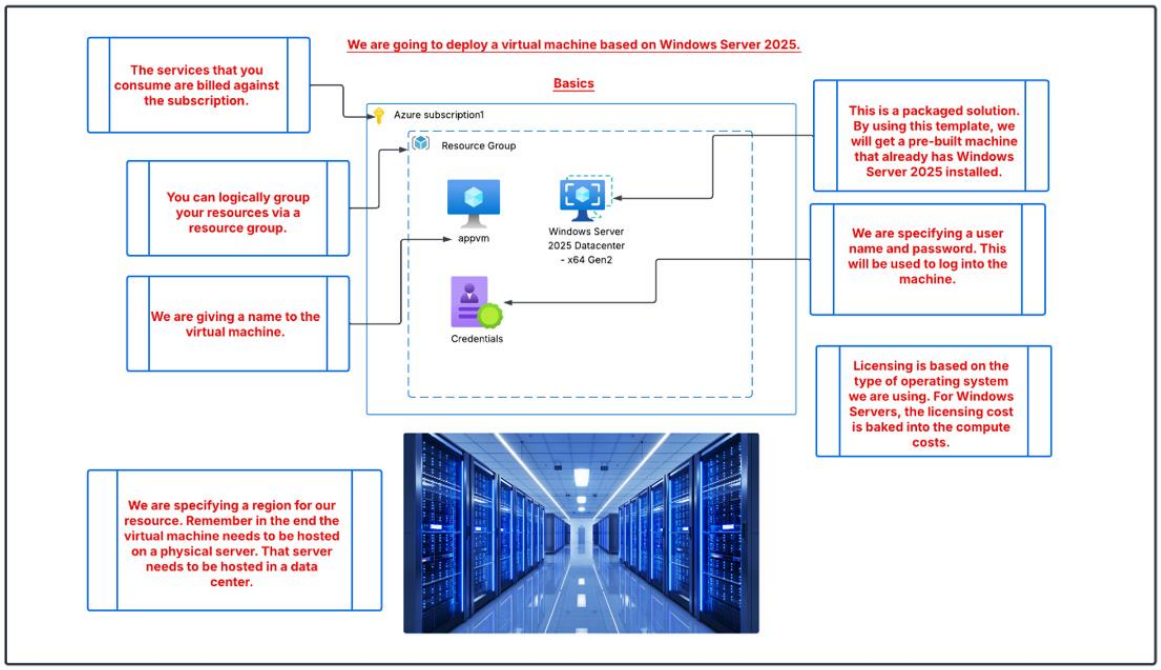
The anatomy behind building an Azure virtual machine



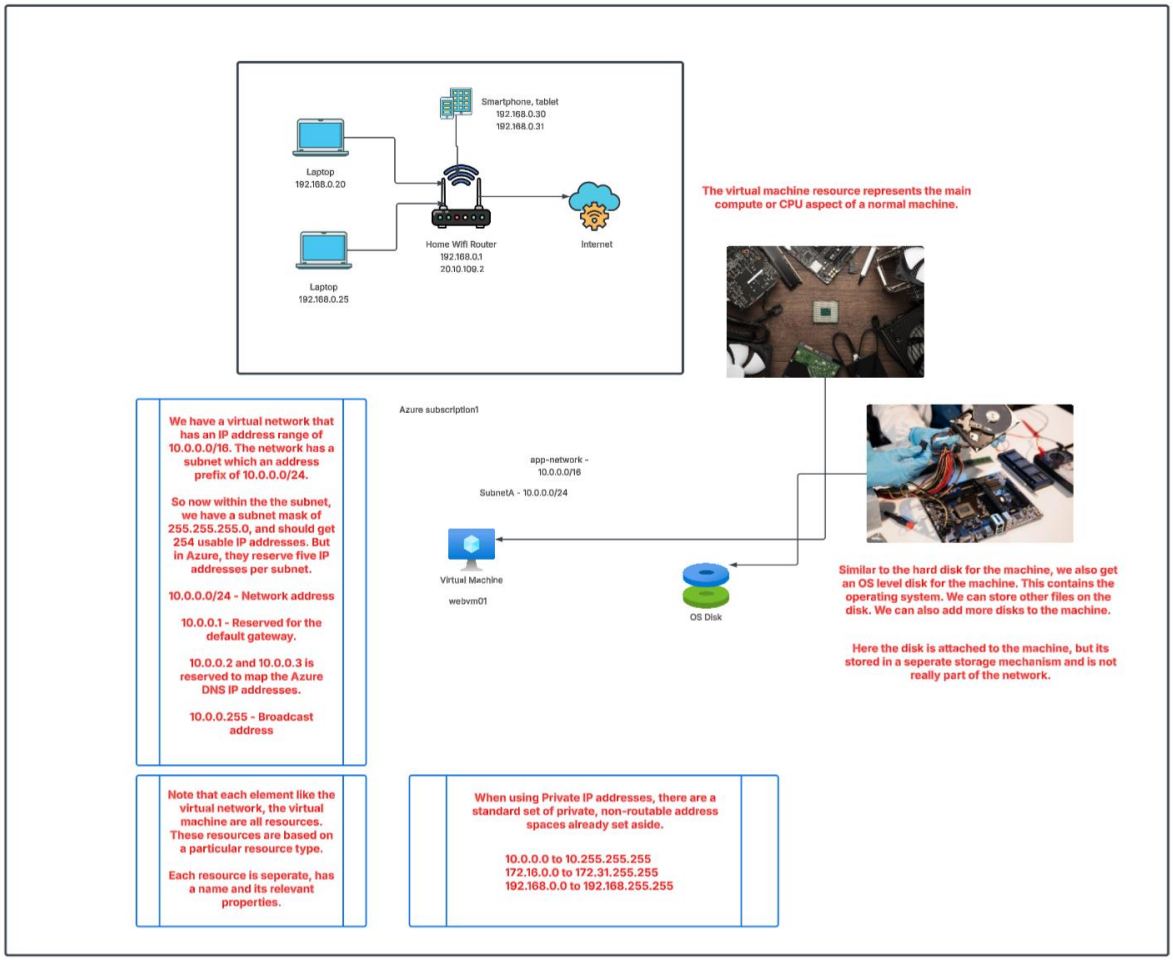


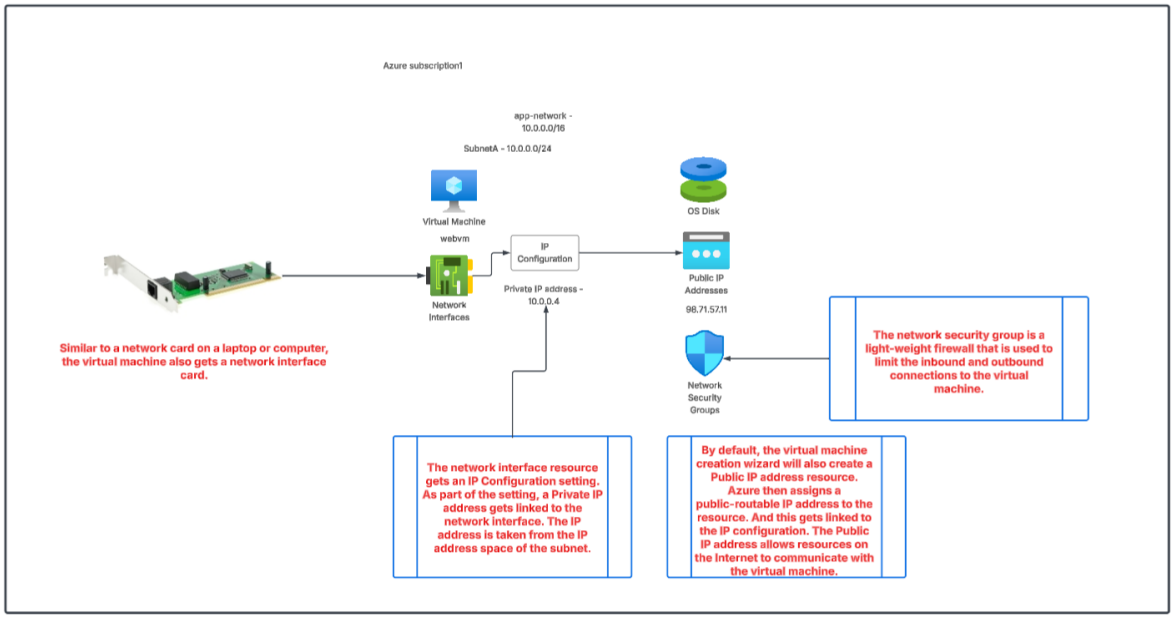
When it comes to the Azure platform, you don't need to buy or maintain the physical infrastructure.

This is managed by Azure. You can just use the service to start deploying virtual machines.

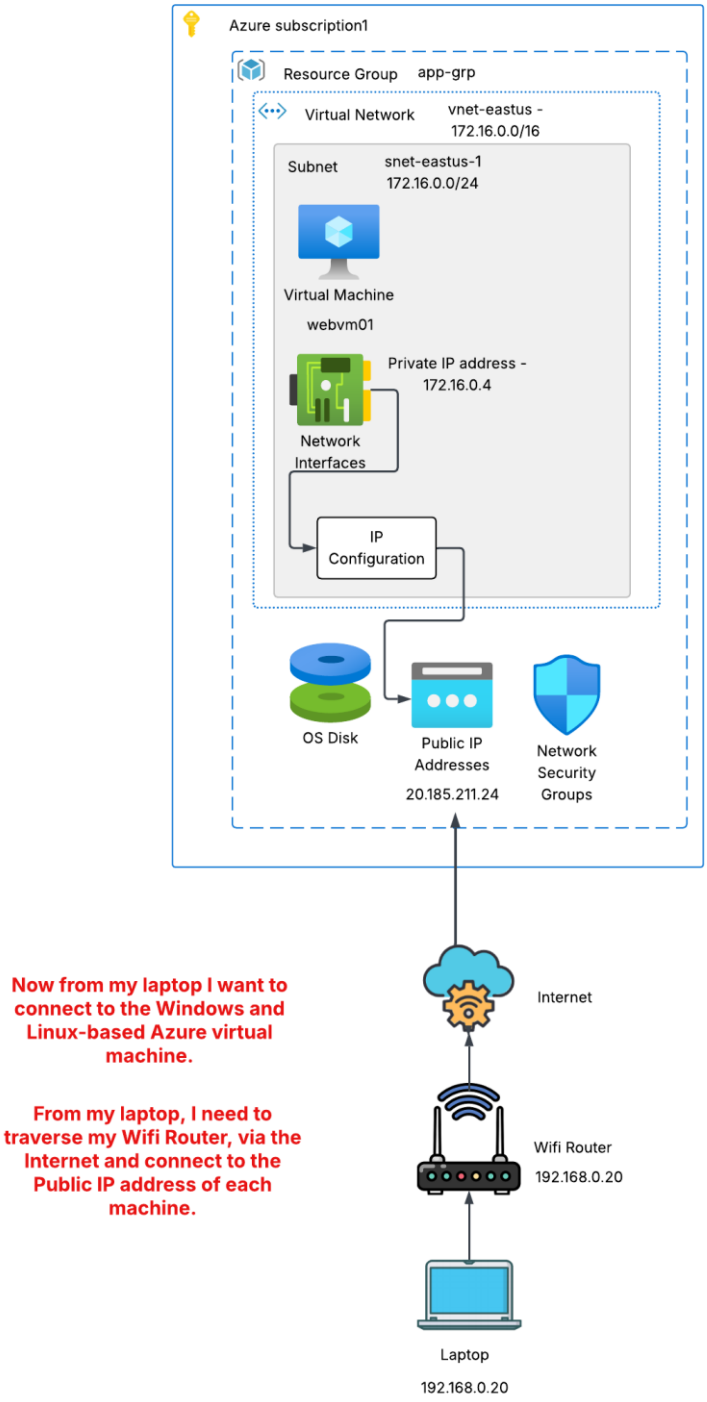


Lab - Building a Windows virtual machine




Azure subscription1
app-network - 10.0.0.0/16
SubnetA - 10.0.0.0/24
Virtual Machine
webvm
Network Interfaces
IP Configuration
Private IP address - 10.0.0.4
OS Disk
Public IP Addresses
98.71.57.11
Network Security Groups
Similar to a network card on a laptop or computer, the virtual machine also gets a network interface card.
The network security group is a light-weight firewall that is used to limit the inbound and outbound connections to the virtual machine.
The network interface resource gets an IP Configuration setting. As part of the setting, a Private IP address gets linked to the network interface. The IP address is taken from the IP address space of the subnet.
By default, the virtual machine creation wizard will also create a Public IP address resource. Azure then assigns a public-routable IP address to the resource. And this gets linked to the IP configuration. The Public IP address allows resources on the Internet to communicate with the virtual machine.

Lab - Connecting to the Virtual Machine



Now from my laptop I want to connect to the Windows and Linux-based Azure virtual machine.

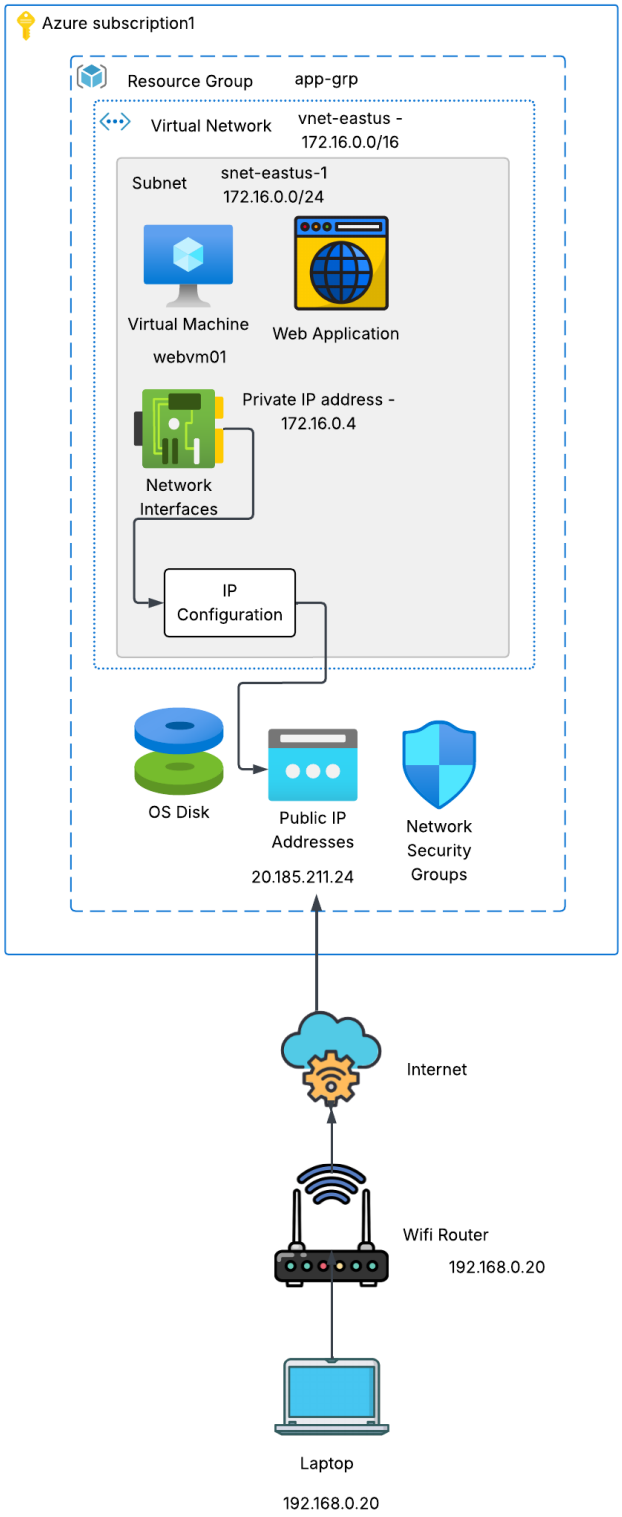
From my laptop, I need to traverse my Wifi Router, via the Internet and connect to the Public IP address of each machine.

Lab - Setting up a Web server on the virtual machine

Let's say that we want to host a web application on the webvm01 machine.

For this we need to install a web server on the machine.

On a Windows Server, we can install a role of a web server. This would install a tool/software known as Internet Information Services.

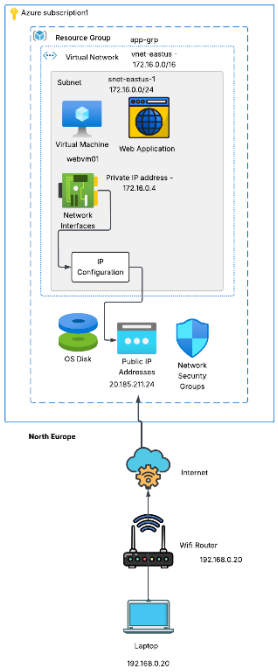


Understanding Azure regions

Azure regions

This is a set of one or more datacenters in a specific geographic location. Here the data centers are connected by a high-capacity, low latency network.

When you deploy resources to Azure, you deploy them to a region. Azure then places the resources on physical infrastructure located in the region.



When you deploy an Azure virtual machine, the machine is deployed to a physical host in an Azure datacenter.

The Managed disks are stored on Azure Storage clusters.

Your virtual network is again part of some physical networking system within the datacenters.

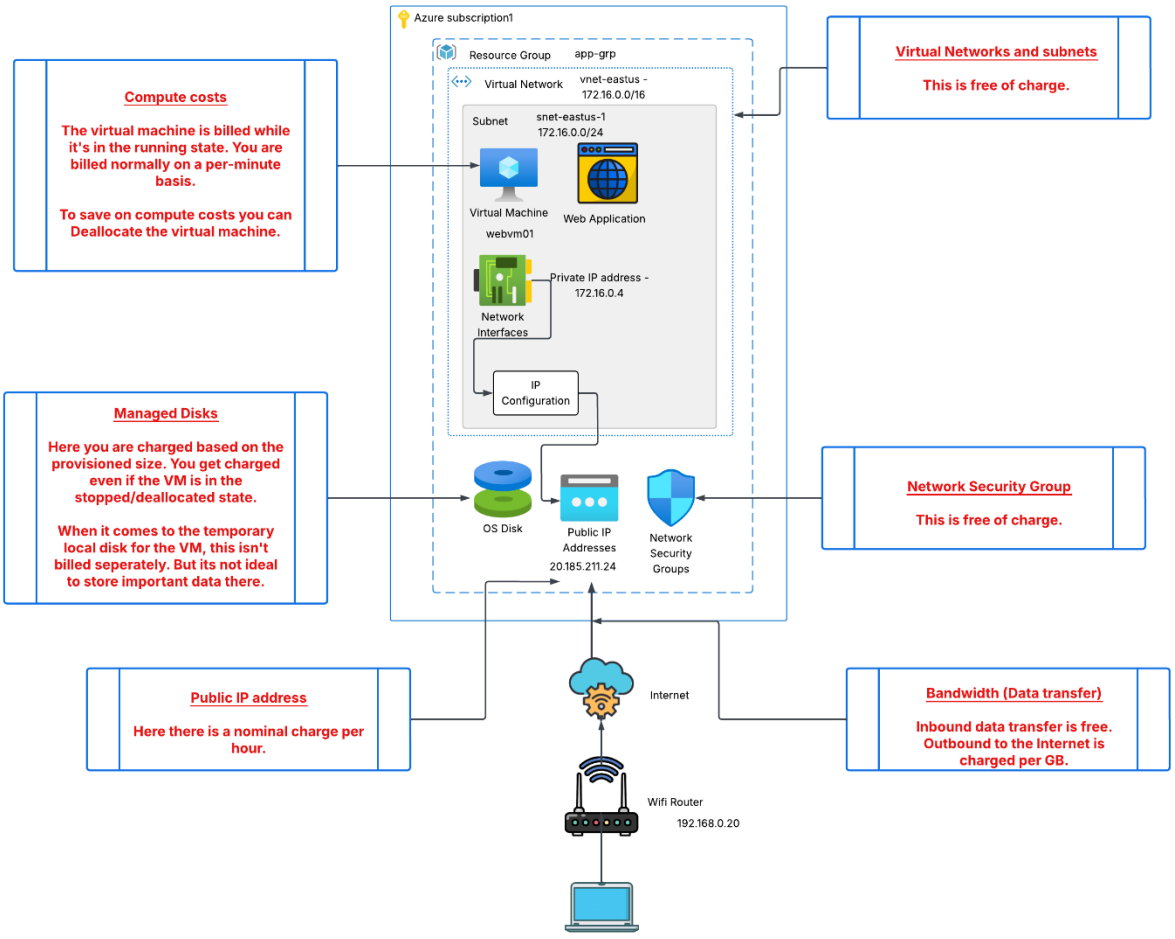
Why choosing a region matters

Latency and User experience - If users are accessing an application hosted on a virtual machine, then having the machine located closest to the users helps to reduce latency and improve user experience.

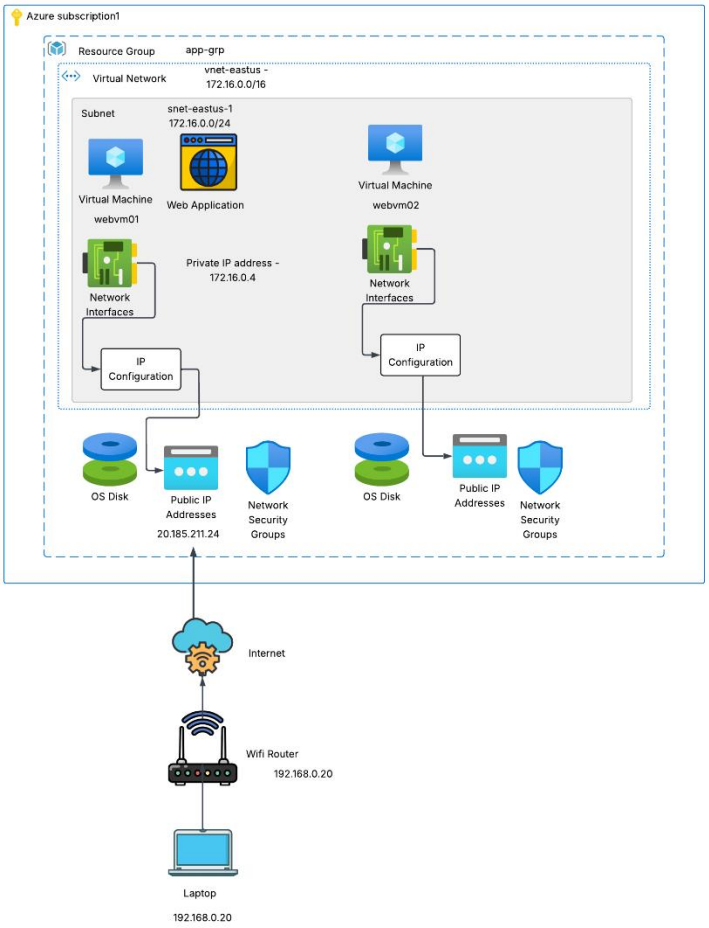
Compliance and data residency - Due to legal and security implications, you might need to host data within particular regions only.

Features and Pricing - Services and SKU's can vary region by region. Prices also differ from region to region. A company might want to cut costs and choose a low-cost region for hosting resources.

Costing for Azure resources



Lab - Building a Linux Virtual Machine



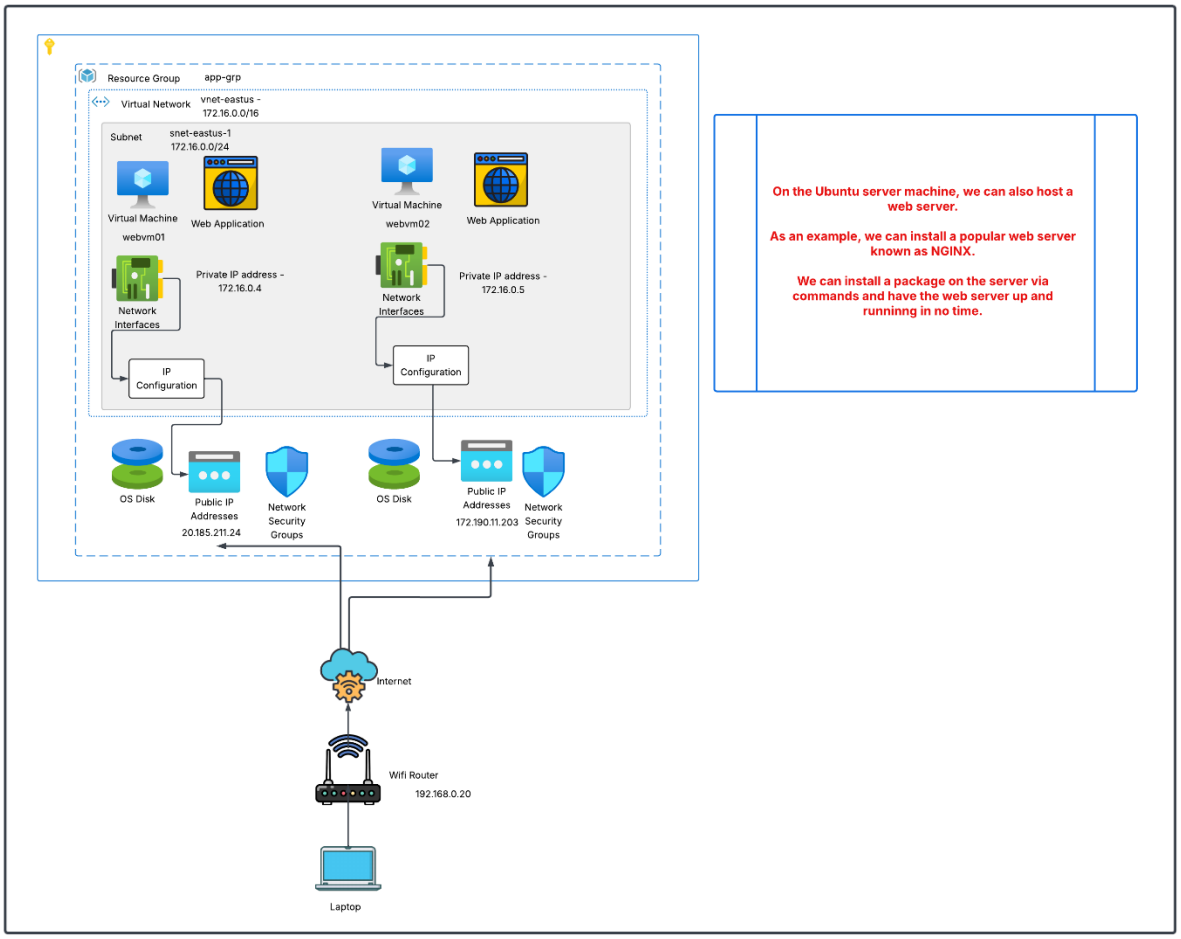
Let's deploy another virtual machine to our subnet in the virtual network.

This machine will be based on a Linux distribution.

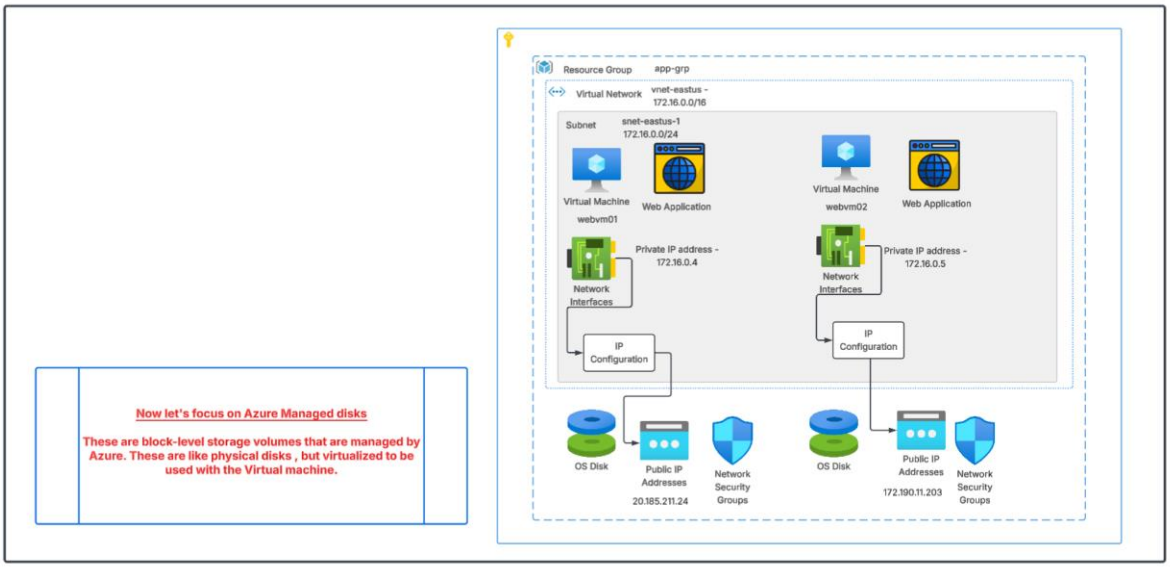
Note that we can deploy the machine to another subnet to provide a security boundary across workloads.

Or if we want to complete isolate the compute machines, we can deploy the machine to another Azure virtual network.

Lab - Deploying a web server on the Linux virtual machine



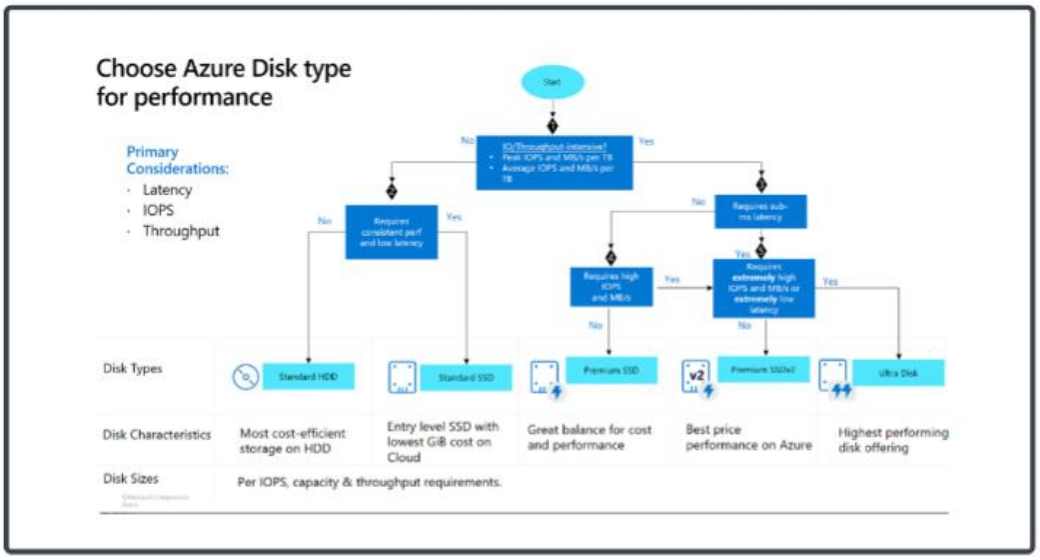
Azure Virtual Machine – Disks



Different types of disks

	Ultra Disk	Premium SSD v2	Premium SSD	Standard SSD	Standard HDD
Disk type	SSD	SSD	SSD	SSD	HDD
Scenario	IO-intensive workloads such as SAP HANA, top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads.	Production and performance-sensitive workloads that consistently require low latency and high IOPS and throughput	Production and performance sensitive workloads	Web servers, lightly used enterprise applications and dev/test	Backup, non-critical, infrequent access
Max disk size	65,536 GiB	65,536 GiB	32,767 GiB	32,767 GiB	32,767 GiB
Max throughput	10,000 MB/s	1,200 MB/s	900 MB/s	750 MB/s	500 MB/s
Max IOPS	400,000	80,000	20,000	6,000	2,000, 3,000*
Usable as OS Disk?	No	No	Yes	Yes	Yes

Reference - https://learn.microsoft.com/en-us/azure/virtual-machines/disks-types



Reference - https://learn.microsoft.com/en-us/azure/virtual-machines/disks-types

Quick note on IOPS and Throughput

IOPS - This is the physical Input and Output operations per second for storage.

SQL
Databases

Let's say you have a SQL database server that hosts a SQL database.

The actual data files for the database tables would reside on the disk.

The CPU is required from a computational aspect for the database server.

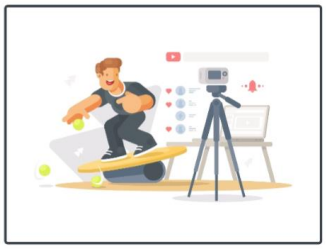
When the user fires a query to search for data in a database table, the database server needs to perform some basic steps

1. First see if the data already resides in memory.
2. If not , the database server needs to read the data from the disk and then load into memory.
3. The data would then be read from memory.

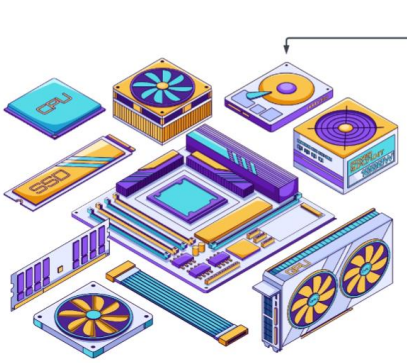
Hence the operation of actually reading data from the disk is an Input operation.

With production databases, with numerous queries, you get a lot of Input and output operations.

Throughput - This is the amount of data per second that the disk can accept or give.

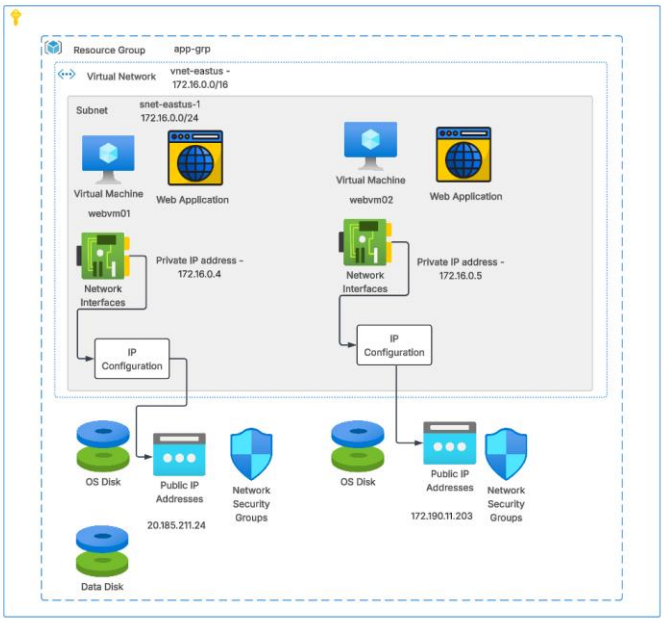


Let's say you have an application hosted on the compute machine that accept videos uploaded by users.



You can have some videos that are larger in size. By having a larger throughput, you have larger chunks of videos transfered at a faster rate.

Lab - Adding data disks



Let's add a data disk to webvm01 machine.

First we need to create the data disk which is an additional resource. Then we attach the data disk to the virtual machine.

Lab - Data Disks Snapshot

Azure Disk Snapshot

A snapshot is a full, read-only copy of a virtual hard disk (VHD).

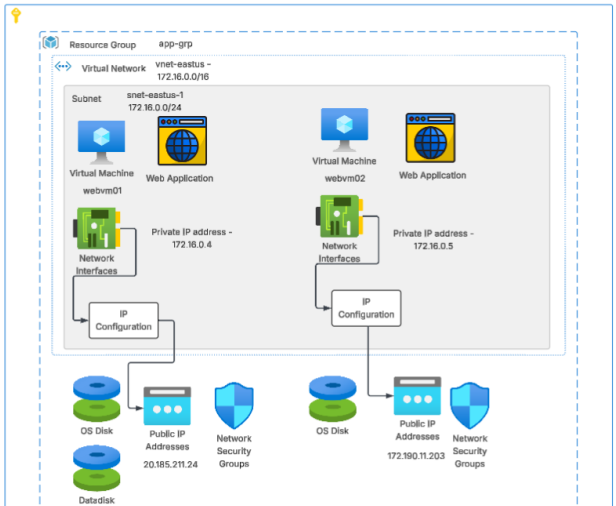
A snapshot can be take of the operating system and the data disks.

The snapshot can act as a point-in-time backup.

Or you can use the snapshot data to bring up a new disk if you need to diagnose or troubleshoot issues.

As a first step, let's ensure we have some data on our data disk.

Then we'll create a disk snapshot from the Azure Portal.

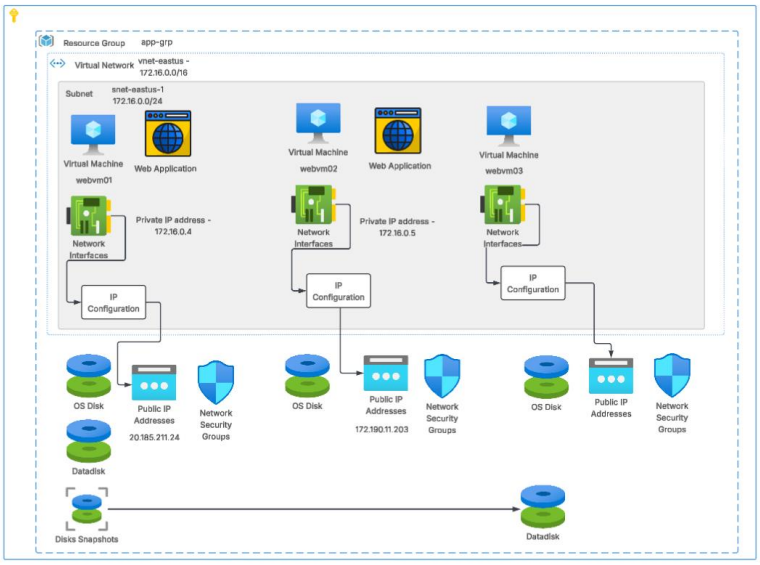


Then in order to attach the disk to another machine, let's create another Azure virtual machine based on Windows Server 2025.

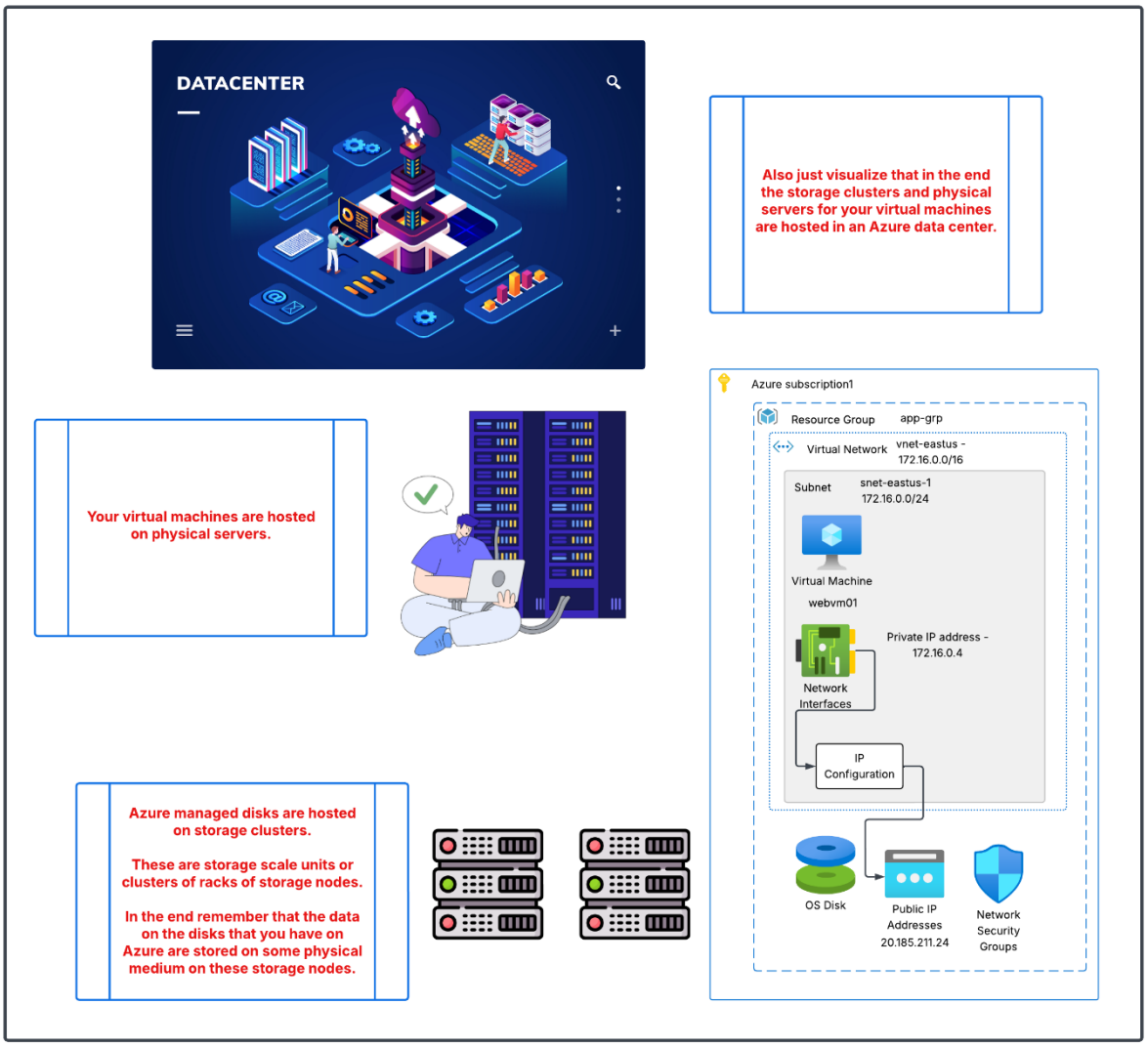
This time, let's create another subnet and add the machine to that subnet.

Then we will create a new data disk from the Disk snapshot.

Attach the new disk to the new virtual machine and confirm that we have data on that disk.



Azure Disks - Server Side Encryption



Now even though the data resides in physical storage in an Azure data center. And Azure data centers are secure by default. Still to make your data secure at rest, there are different storage encryption options.

Server-side encryption of Azure Disk Storage

This is enabled by default. This encrypts the data stored on Azure managed disks - both OS and data disks.

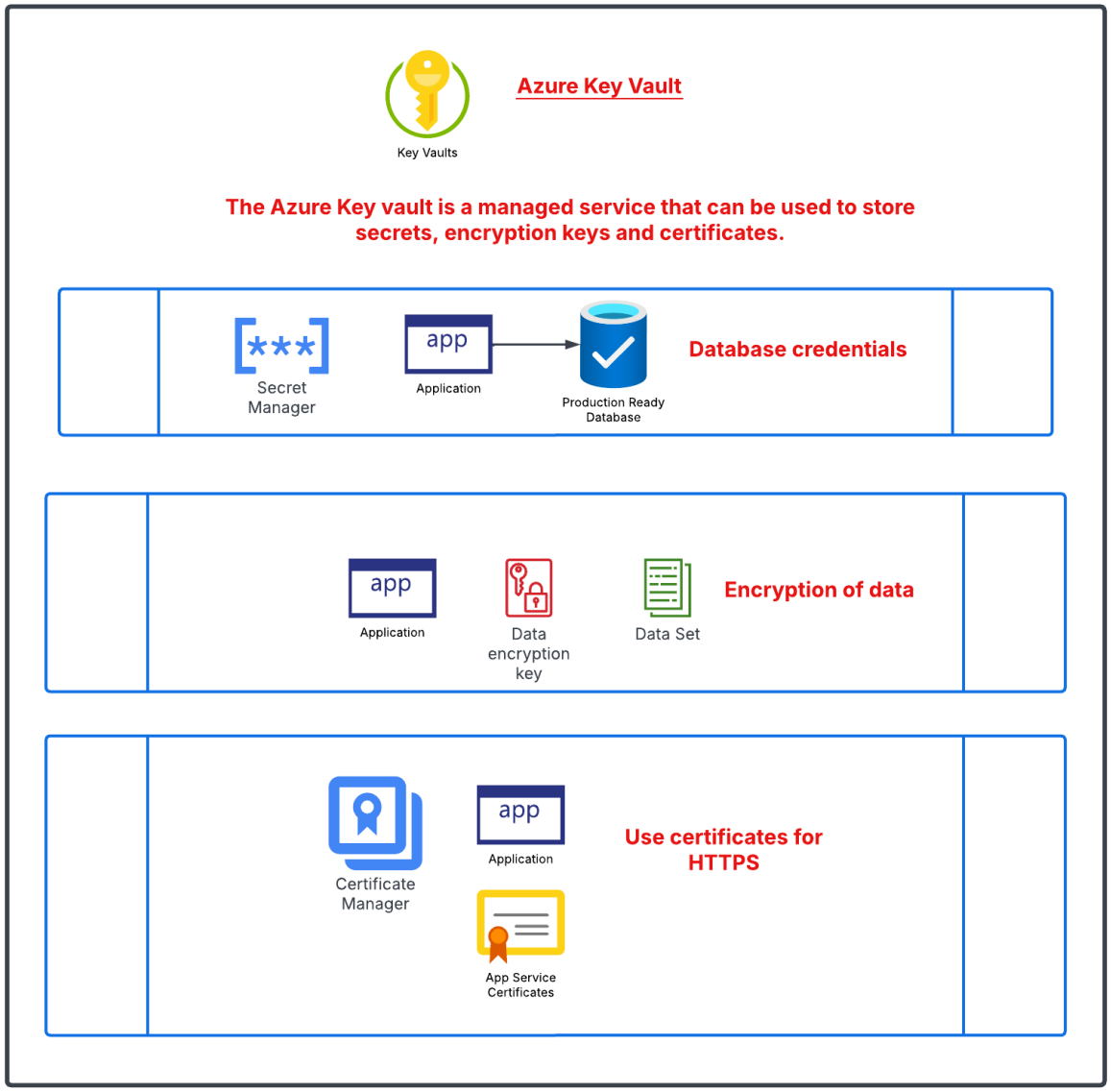
Here Azure manages the encryption. Here data is encrypted using an encryption algorithm and an encryption key.

This does not impact the performance of managed disks and there is no additional cost.

By default Azure manages the encryption keys - These are known as platform keys.

But you can also make use of customer-managed keys - Here you can specify the encryption keys used to encrypt the disks.

Lab - Azure Key Vault Service

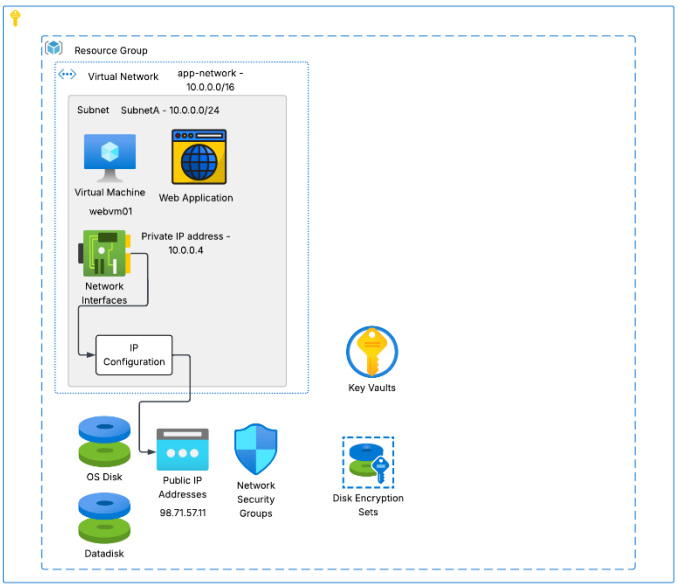


Lab - Disk Encryption Sets

1. Create an encryption key in the key vault. Only software and HSM RSA keys of sizes 2,048-bit, 3,072-bit and 4,096 bit are supported.

2. The Azure VM, disks and key vault need to be in the same subscription and the region.

3. Next we need to create a disk encryption set. This would contain the key from the key vault.



Custom Script Extensions

We had seen earlier on an example of setting up a workload on an Azure virtual machine. On Windows, to setup a web server, we could install Internet Information Services.

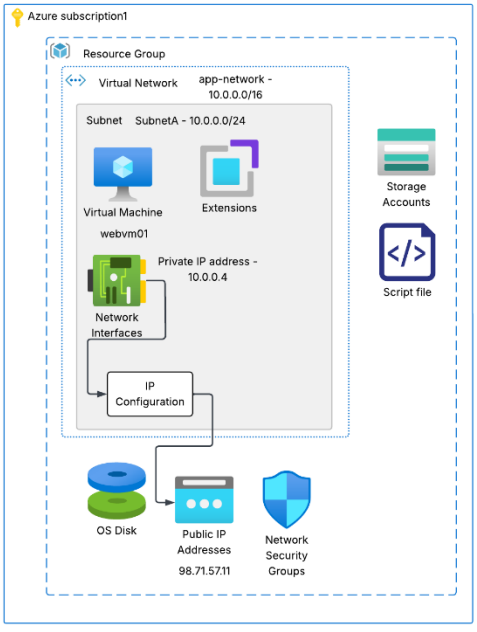
But we can automate the task of installing the web server directly after the deployment of the virtual machine. No need to manually log onto the machine and install the web server.

With the help of Custom Script Extensions, the extension will download and run the script.

This is ideal for post-deployment configuration, software installation tasks etc.

You need to develop the script that would perform the required task - like installing Internet Information service.

When developing the script, try not to develop scripts that require user inputs or restarts. These scripts would run in the background and would run once. The script would also be allowed 90 minutes to run.

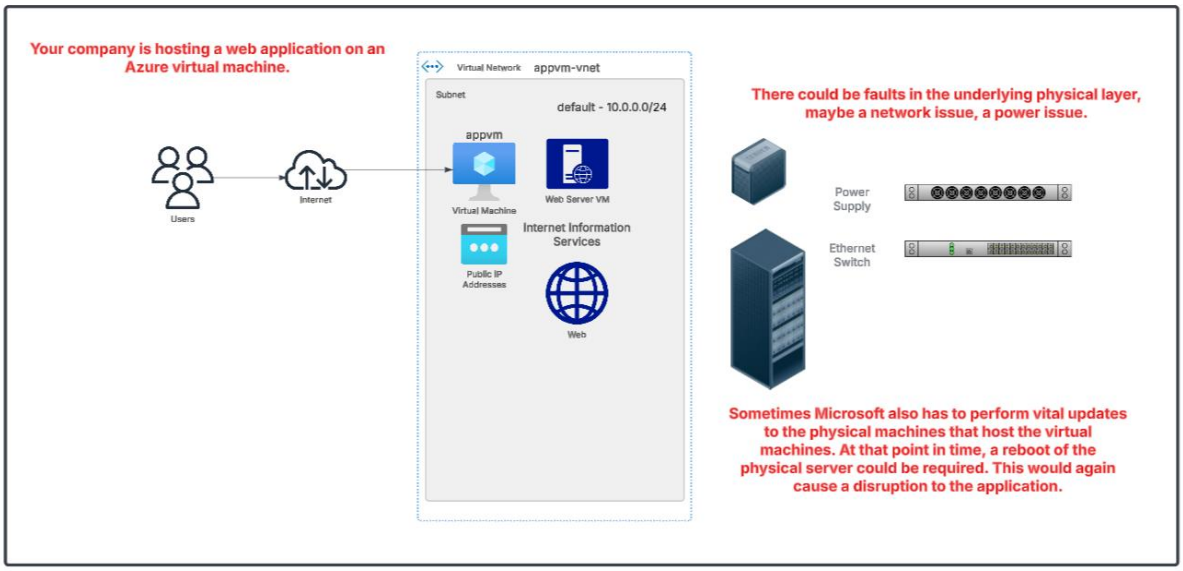


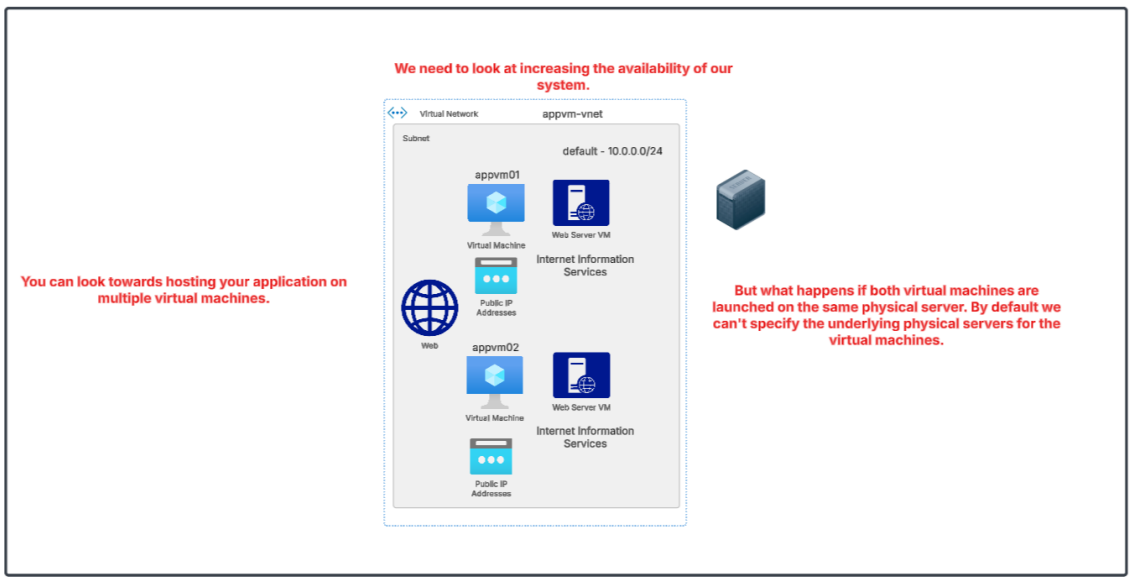
We need to supply the script which can reside in an Azure storage account, GitHub or any public URL which would allow download of the script.

We will make use of Azure Storage accounts. We will first create an Azure storage account. Then upload a script that would automatically setup Internet Information Services.

Then we will launch a new Azure virtual machine based on Windows and specify that we need to use Custom Script Extensions.

Availability Sets



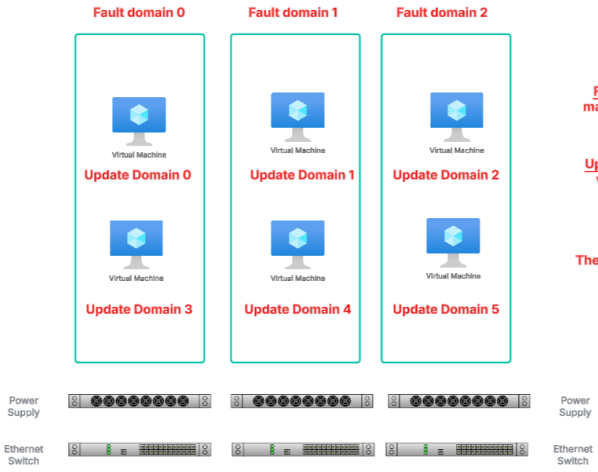


Availability sets

This is a logical grouping of machines that helps to reduce the chances of multiple VM's going down because of hardware issues.

To make use of Availability sets, you need to deploy a Virtual Machine to an Availability Set. You can just create an Availability Set and deploy the machine to the set. The machine can only be part of a set when the machine is created.

The virtual machine is placed as part of a fault and update domain in the Availability set.

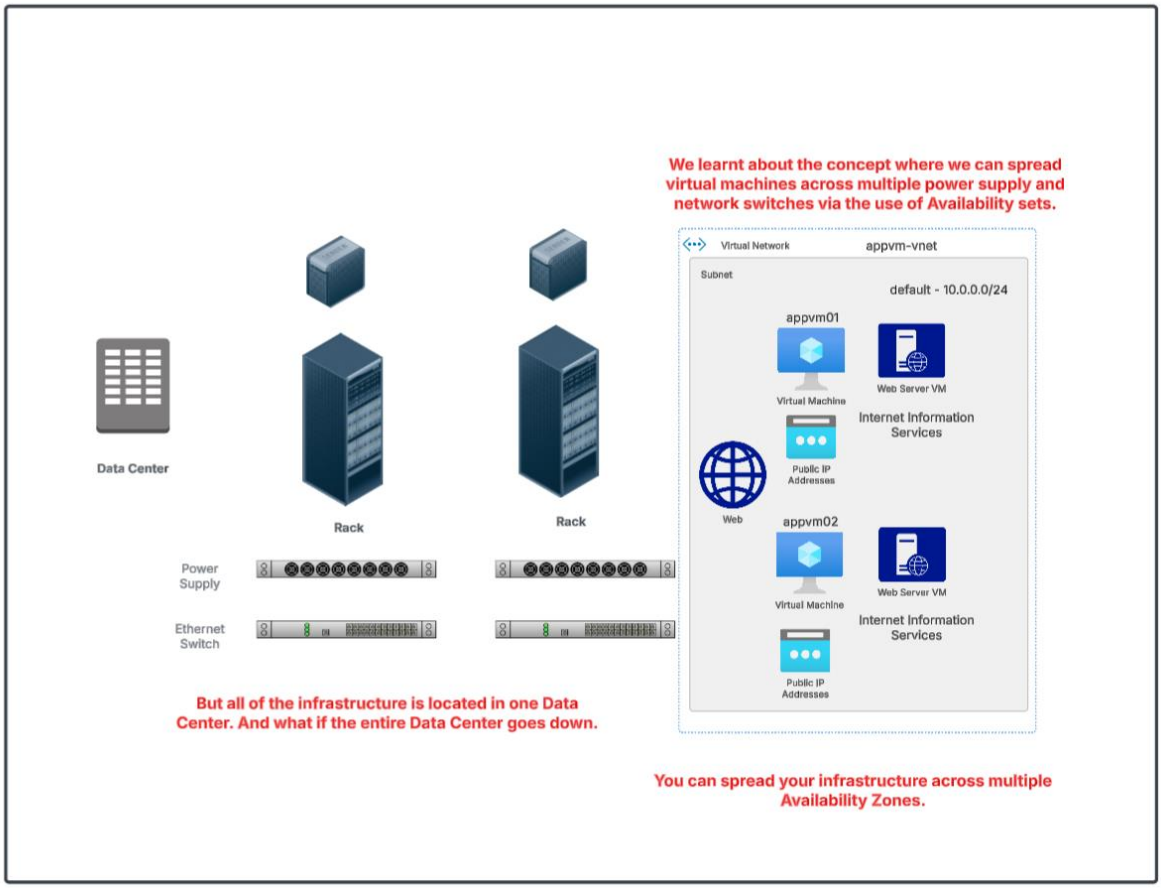


Fault domains - This helps to define a group of machines that share a common power source and network switch.

Update domains - This helps to signify the group of virtual machines and physical hosts that can be rebooted at the same time.

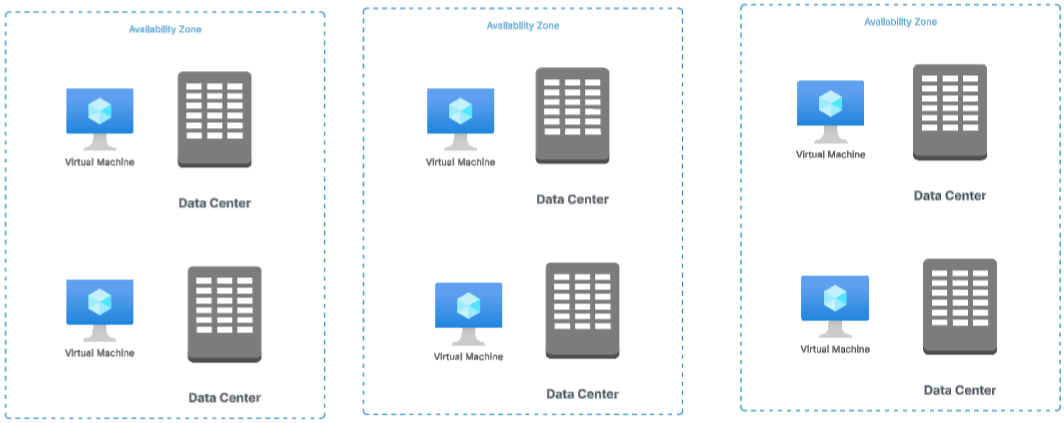
There is no additional cost for using Availability Sets.

Availability Zones



An Availability zone is a group of data centers. There are fast links across Availability Zones to ensure low latency.

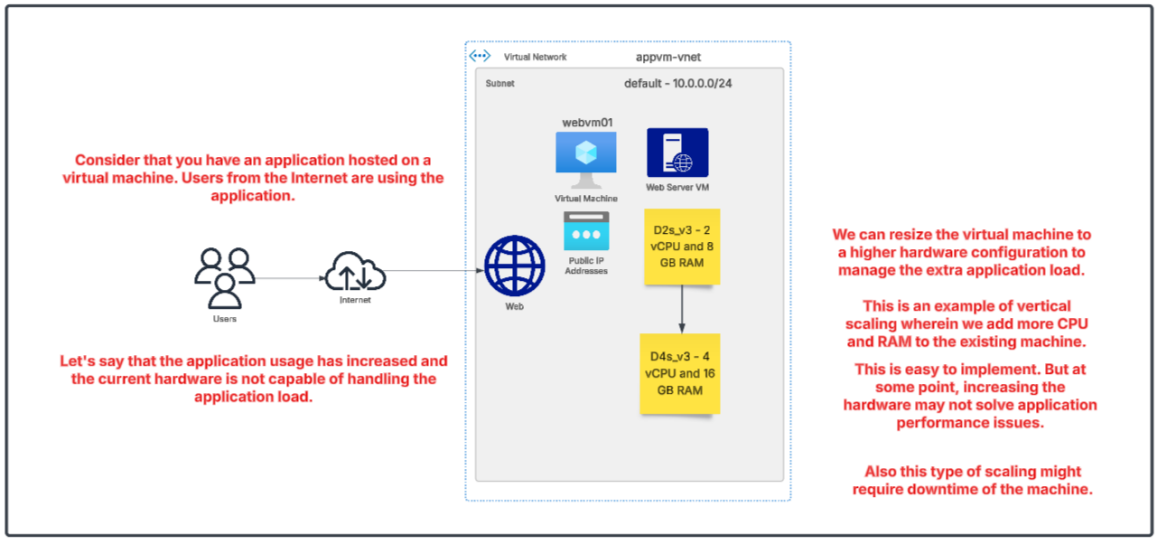
An Azure region has multiple Availability zones.

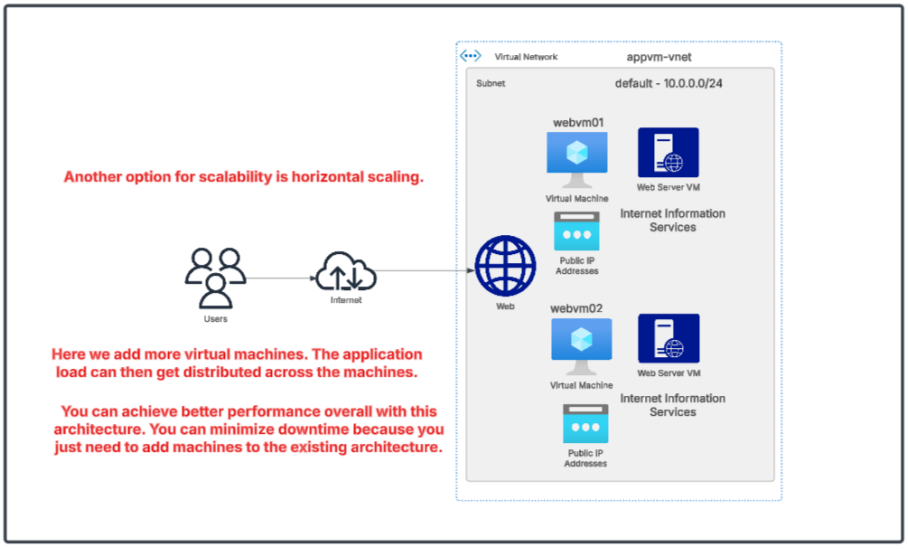


Data Transfer	Price
Data Transfer In	Free
Data transfer between Availability Zones(Egress and Ingress)*	$0.01 per GB
Data transfer within same Availability Zone	Free

There is no cost for using an Availability zone, but there is a bandwidth cost.

Azure virtual machine scale sets





How do we manage the adding of machines to our architecture?

Maybe the traffic is high during the day time and light during the night time and we don't need so many machines during the night. Why does this concern us? Well the cost of having machines in place.

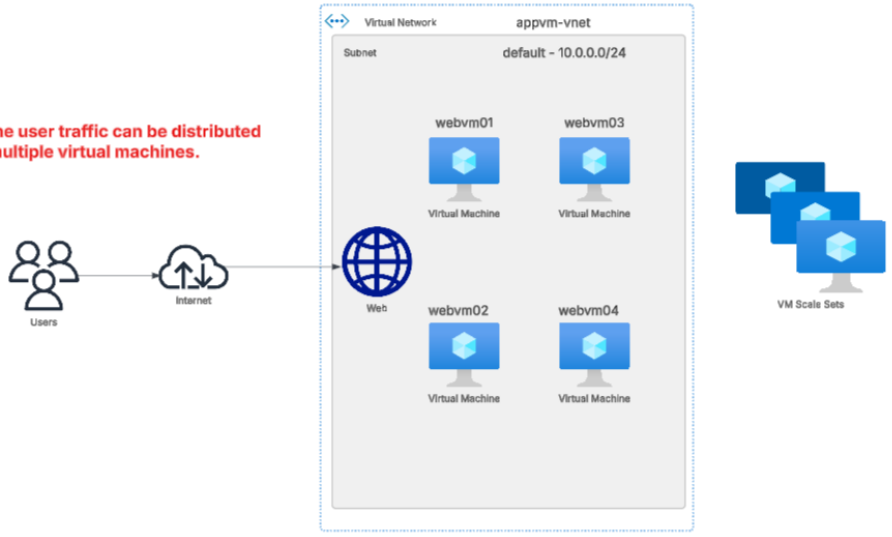
Are you manually going to monitor whether the traffic is high or low, and create and delete machines whenever required.

Azure virtual machine scale sets

This service helps you to create and manage a group of load balanced virtual machines.

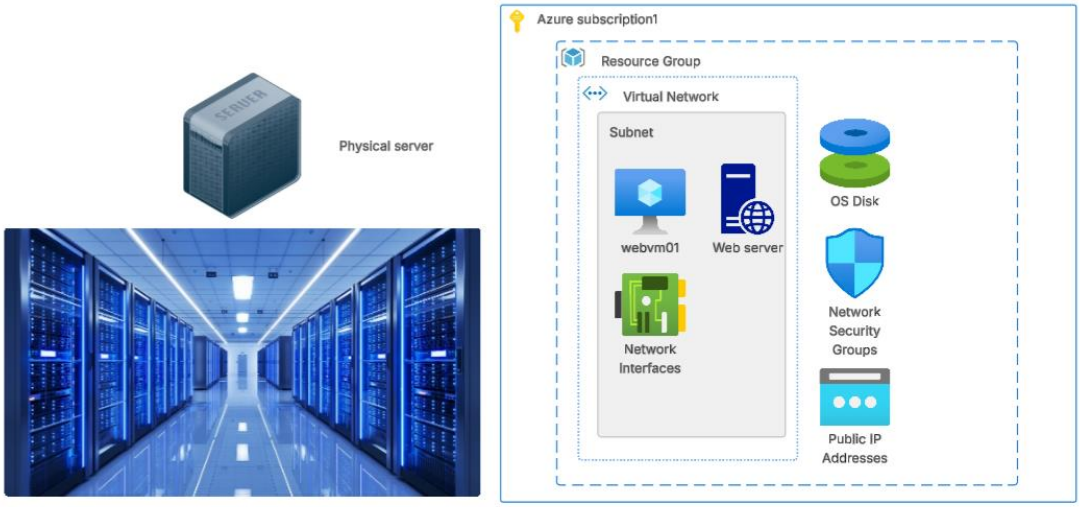
The number of virtual machines can then grow based on demand or on a schedule.

By Scalability, the user traffic can be distributed across multiple virtual machines.



Your machines can be managed by the virtual machine scale set service. You can define rules to define different scaling conditions.

Introduction onto Azure Web Apps



We can host web applications on an Azure virtual machine. We can install a web server and setup our application accordingly.

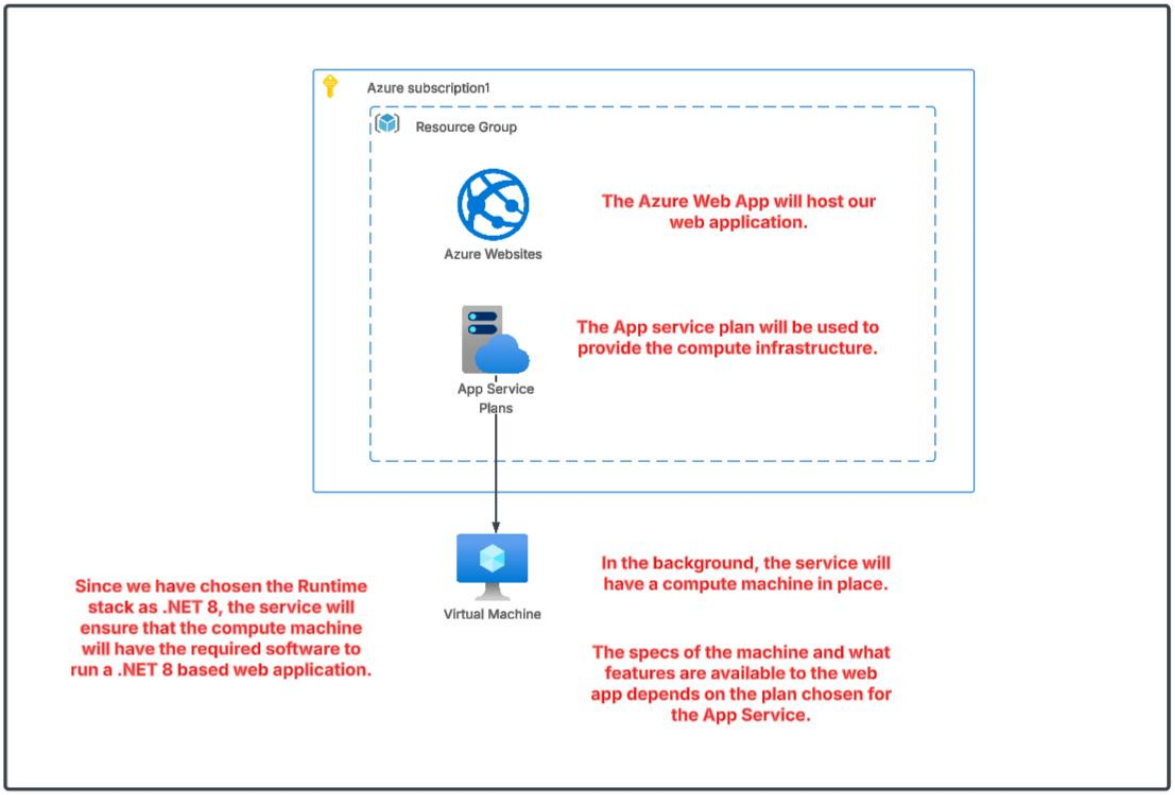
When it comes to a virtual machine, the under physical server is managed for you in an underlying data center.

But maybe the company who wants to host the web application does not want the hassle to even manage the virtual machine infrastructure.

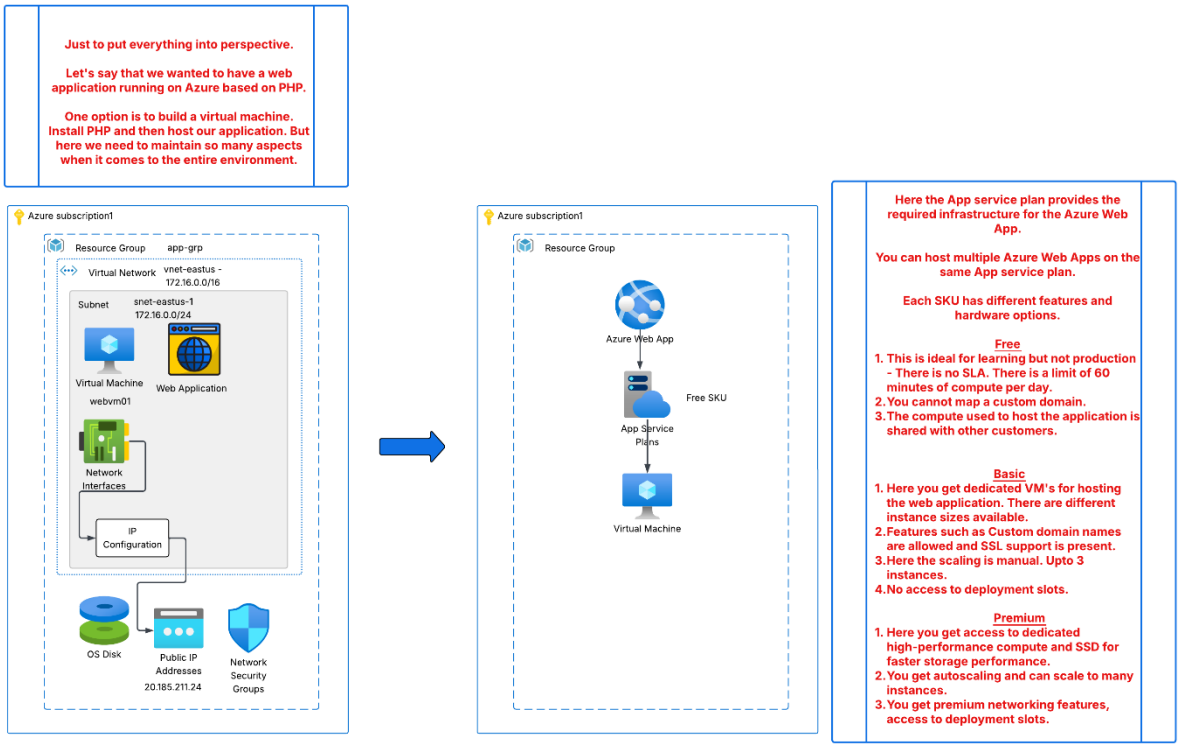
They can opt to use the Azure Web App service.

If you have a web application that fits the framework and you don't want to manage the virtual machines, then you can opt for the Azure Web App service.

But if you need to host a custom application that needs to be installed, then you would probably need to use the Azure virtual machine service.



More on App Service Plans



Azure Web Apps - Deployment slots

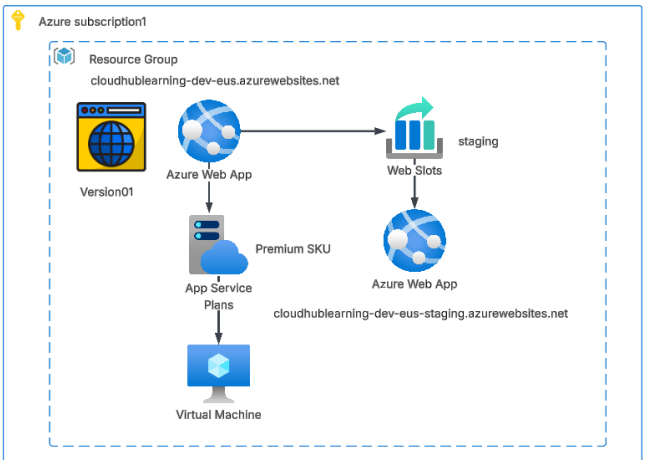
Now normally applications, such as your web applications undergo a lot of changes. You can have new features added or bug fixes added to code.

So your application can go through multiple revisions.

But sometimes these changes could break your application. The application stops working in productions, users are not happy.

For Azure Web Apps, there is a feature known as deployment slots. This is like an extra live site for your web application.

Let's understand this based on an example we are going to implement.

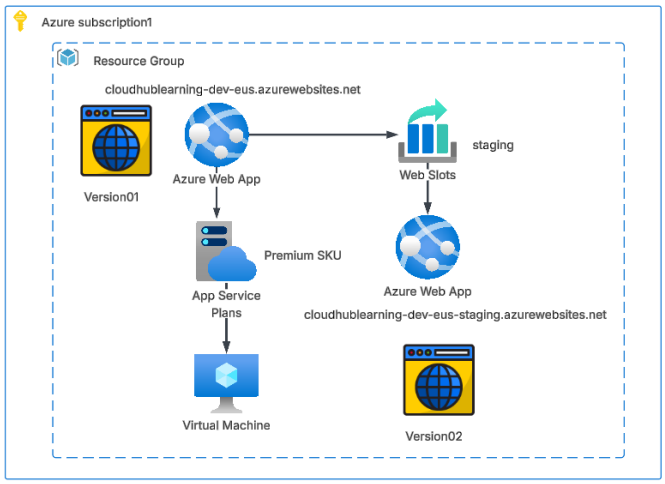


Let's assume that we have Version 1 of our application running on our Azure Web App resource.

Firstly in order to use deployment slots, we will need to bump up the SKU of our App service plan. We need to choose the Standard App service plan or higher.

Based on this our machine now becomes dedicated compute - Currently we would get a machine with 1 vCPU, 4 GB of RAM and 250 GB of storage

Next we would create a new deployment slot.

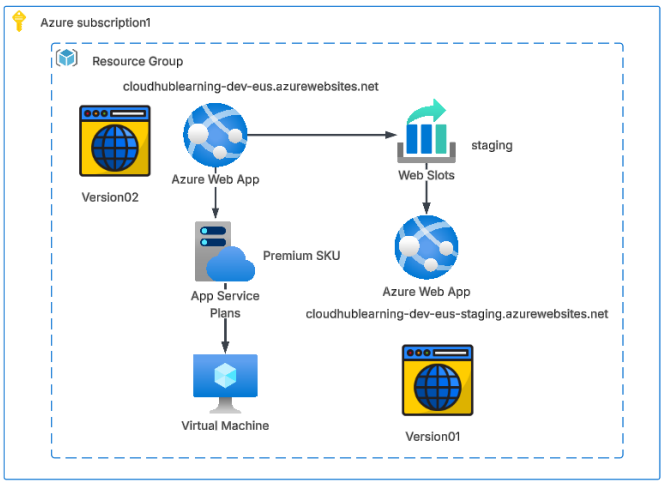


Now for the application running in the staging slot, we can upload the newer version of our code.

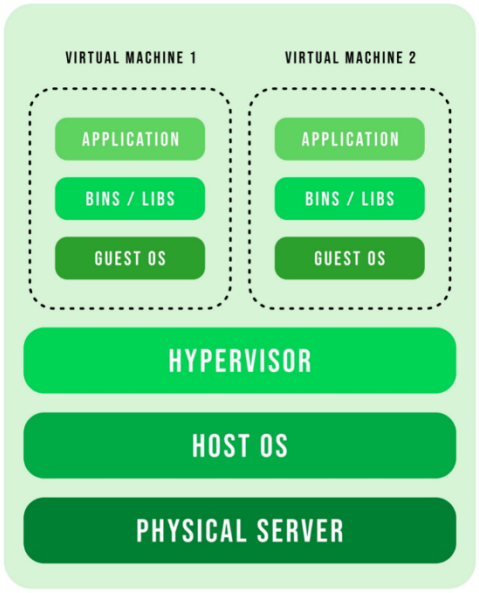
Users can then test the newer code version in its own environment now.

Once the testing is complete and signed off, we can push the new code base to our production slot by just performing something known as a swap operation.

After the swap



Container-based applications



VIRTUAL MACHINE
ARCHITECTURE

Virtual Machines were designed to make the most of the hardware specifications of the underlying physical machine.

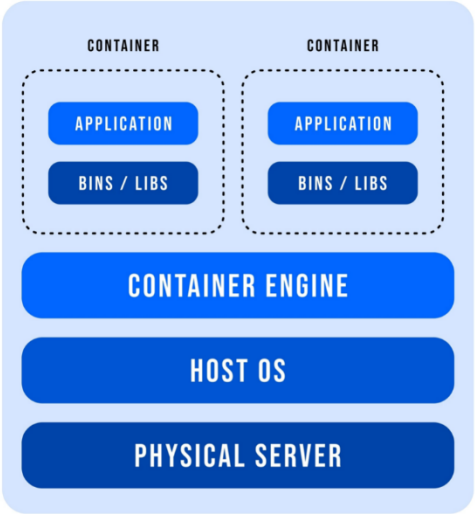
You could have multiple virtual machines hosted on the same physical server.

But then there were still some problems with it came to application deployments.

Companies would deploy multiple applications on the same virtual machine. They want to make the most of each virtual machine.

But let's say that one application requires an update at the virtual machine operating system level. This update could break the other applications running on the same virtual machine.

The other most common issue was when applications were copied from one environment to another - From development to production. The application would be perfectly working on development but would not work on production due to configuration differences.



CONTAINER
ARCHITECTURE

Then came the advent of container-based tools such as Docker. These tools allowed you to package your code and all dependencies required by the application within a container.

That container could the run on any machine that had the underlying supported container runtime installed such as Docker.

Each container can run independently.

You can install updates for each seperate container as required by the application and it would not affect other containers running on the same host system.

Docker is a popular container platform that provides tools to
1) Package applications into Docker images
2) Runs the images as Docker containers

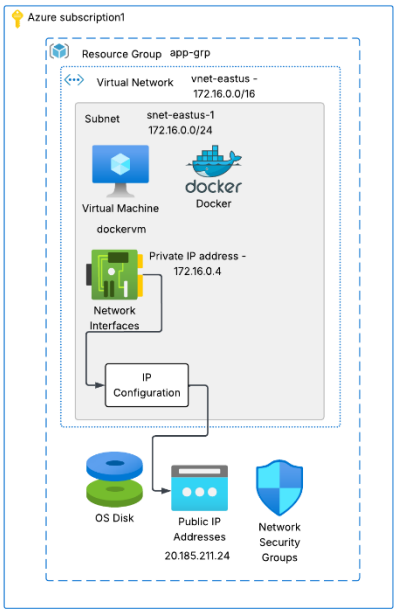
Lab - Setting up Docker on an Azure virtual machine

Now we need to see the different services available on Azure that will allow us to host container-based applications. We will cover just the services based on the exam objectives.

We will target to take a simple PHP application. Package the application into a Docker image. And then run that image as a container.

First and foremost, let's focus on how we can create Docker images and run containers out of them.

For this we will first create an Azure VM based on Ubuntu server. On that server we will install the Docker engine.

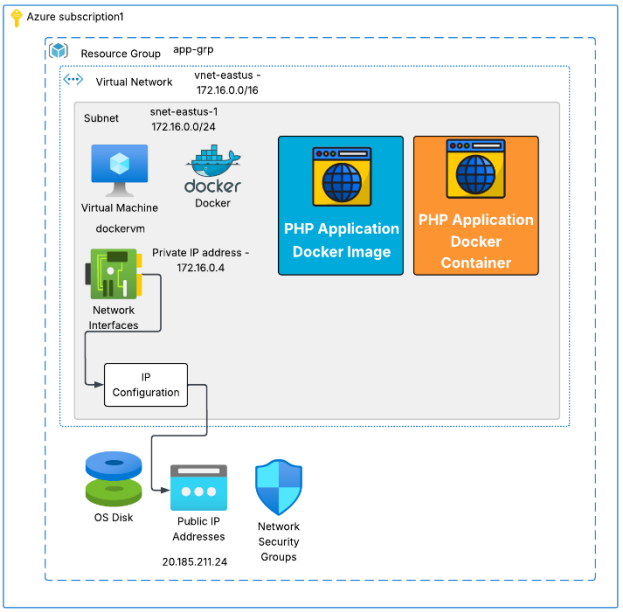


Next we will copy our PHP application files onto the Azure virtual machine.

We will then use the Docker toolset to create a Docker image.

Then from the Docker image we can run a container.

We can then access our PHP application via the Docker container.



Lab - Azure Container Registry service

Now we are able to run an application within a Docker container via the use of a Docker image on the Azure virtual machine.

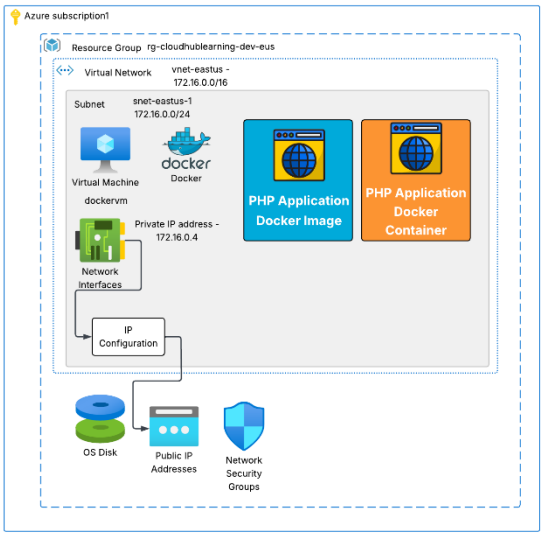
Now let's say that we need to run the Docker container on another machine.

Well , we would need to install the Docker engine on the target machine. Then copy the Docker image to the target machine and then run the container.

For the moment let's target the aspect of needing to copy Docker images from one machine to another. Not the most safest approach to sharing Docker images.

Well Docker as a company has a public container image registry known as Docker Hub. This hosts millions of images and also includes a host of official images for Linux, databases, programming languages etc.

Developers can push their application images onto Docker Hub. Then from the target machine, they can use the Docker tool set to download the Docker image and run the image as a container.



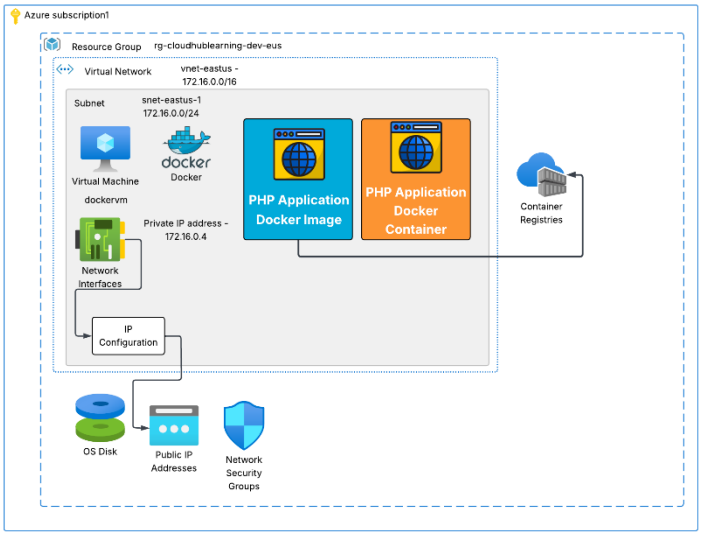
In Azure we have a service known as the Azure Container Registry service.

This is a managed , private Docker registry.

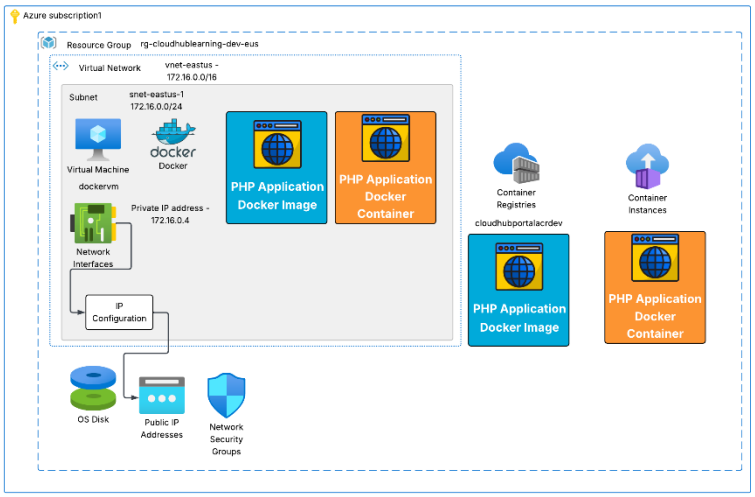
We can push our container images onto this registry.

Key features

1. A private storage for your images
2. Ability to integrate with Microsoft Entra ID for security
3. Network security



Lab - Azure Container Instances



So now we have our Docker image in the Azure container registry.

Now we can run containers elsewhere based on the same image.

As an example of running an image on another service, let's look into Azure Container instances.

Azure Container instances are a serverless runtime environment for containers. Here you don't need to provision compute, download and install Docker. Everything is managed by the service.

You just tell Azure Container Instance where your image is, it will download and run the image as a container.

This is an ideal service to just get a container or set of containers up and running.

Note that if you want complete orchestration of your containers - Autoscaling, rolling deployments, then look into other tools like Azure Container Apps or Kubernetes.

Building a multi-container application

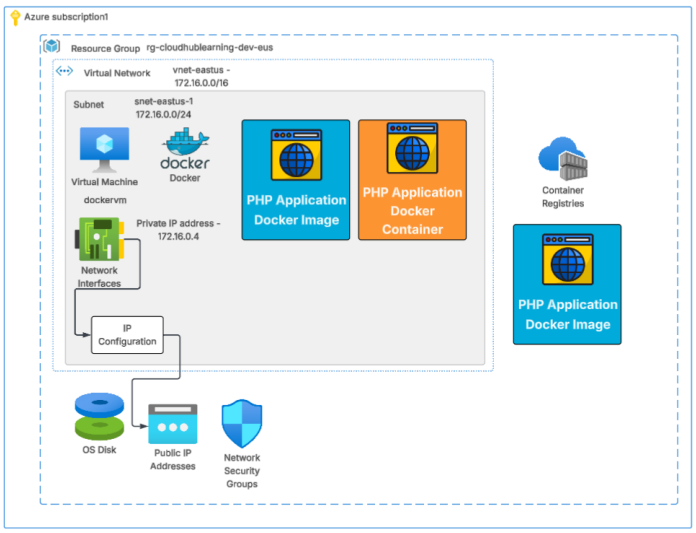
We have looked at building an application-based Docker image and container.

Let's look at a simple example of building a multi-container application. This is so that we understand how to deploy them via Azure-based services.

We will have 2 containers - One for the PHP Application. And we will introduce a new container that will host a MySQL database.

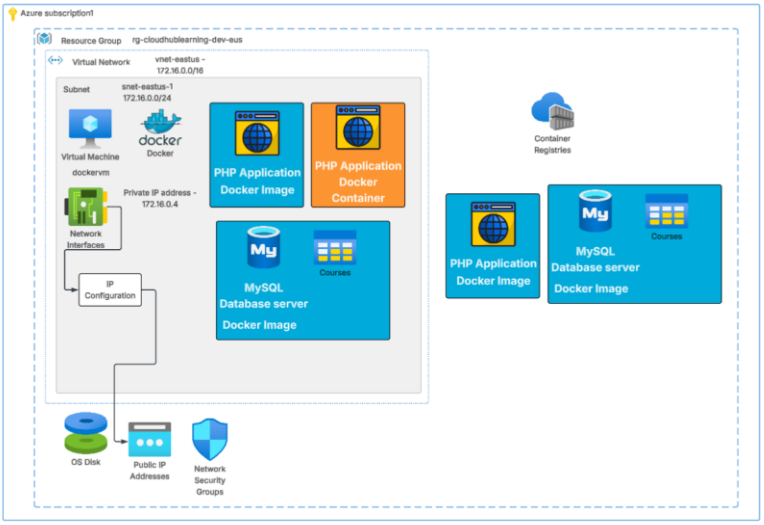
MySQL would run in its own container. It would host a simple database with a table. The table would have information about courses - The Course ID, Course Name and Rating.

This data would be read by the PHP Application running in its own container.



Let's follow the below steps for our entire demo

1. Let's build a Docker Image - This image would take the base MySQL Docker image. With the base image, we would get the MySQL database engine running within the container.
2. We will seed the database with a table to manage the information about courses. This will be part of the Docker image.
3. We will push our new Docker image to the Azure Container Registry.
4. Then just to test the waters, let's deploy the MySQL container to an Azure Container instance to see the data.



Azure Container Apps

Now we have seen how to run a container based on an image using Azure Container Instances - The simplest way to run a container without the need of provisioning the underlying infrastructure.

But what happens when you have multiple containers to manage as part of your application architecture.

Web Frontend Container	Database Container	API Container	Caching Container
This is responsible for serving the frontend Web UI to users.	This is responsible for hosting the backend database.	This container exposes API's that call other functions	This container could be running Redis, which is a popular Caching tool.

In some cases we would need to run multiple instances of a container to take in the user requests.

When you have several containers that need to run as part of your application. And when a company has many such container-based applications, it becomes important to have a software/tool that can manage the container-based applications.

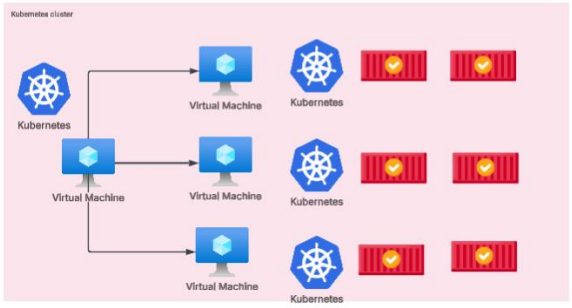
One such popular tool is Kubernetes. This is an open-source container orchestration tool.

In a nutshell, you would setup a cluster of machines. You would install the Kubernetes software on each machine.

One machine would become the Kubernetes controller that would server as the control plane. The other machines would serve as nodes in the cluster. The containers would then run on the nodes.

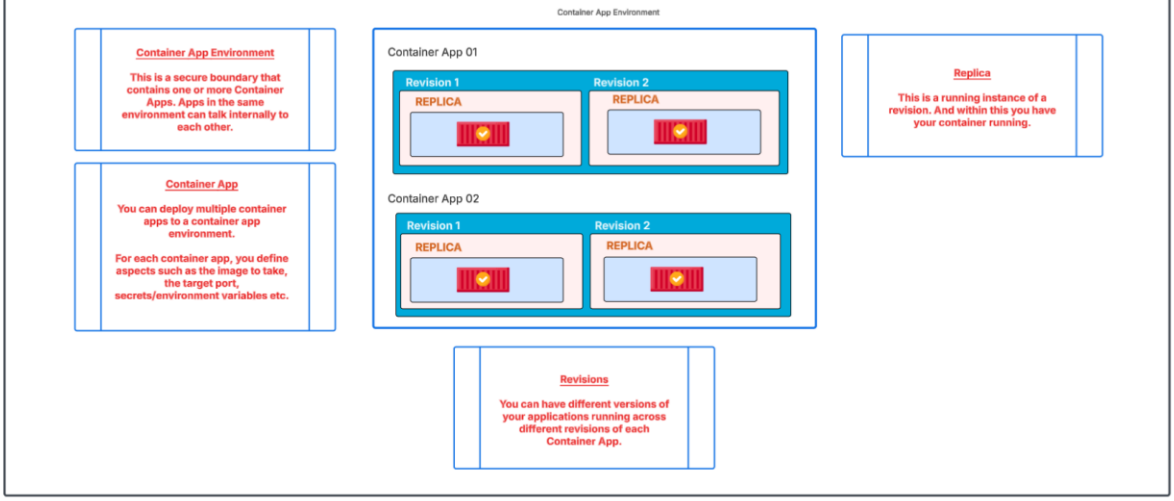
Kubernetes offered several features that included

1. Schedule and placement - Kubernetes would take care of choosing which node would run the container.
2. Self-healing - If a container crashed, it would reschedule the container on a healthy node.
3. Scaling - It could create replicas , basically scaling on demand.
4. Networking - There is support for routing of traffic
5. Rollback - You could implement different patterns like rolling deployments and the rollback updates.



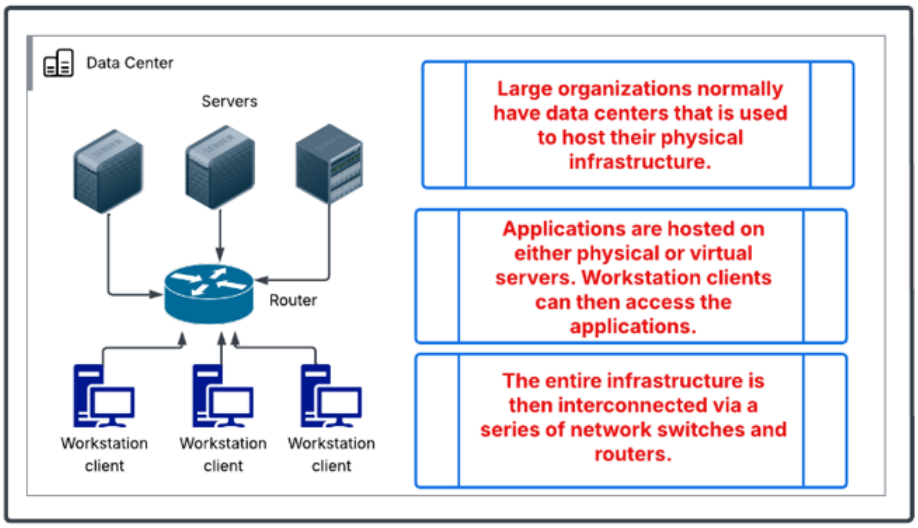
So now what is Azure Container Apps

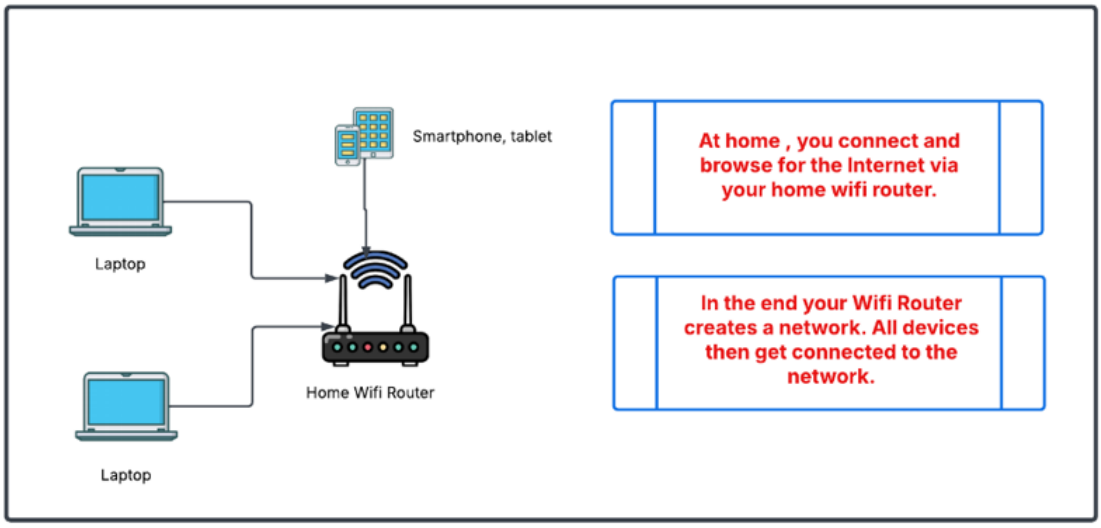
Well this is an abstraction layer built on top of Kubernetes and other open-source tools. Here you don't need to manage the underlying Kubernetes clusters, but still get the benefits of using Kubernetes.

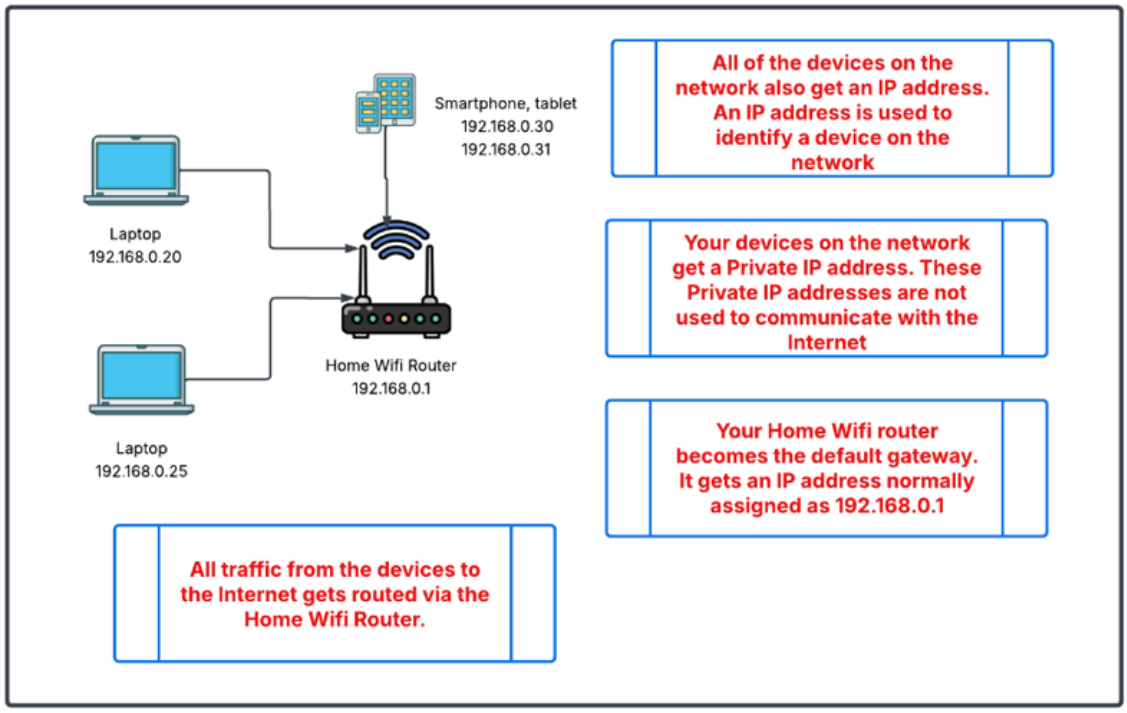


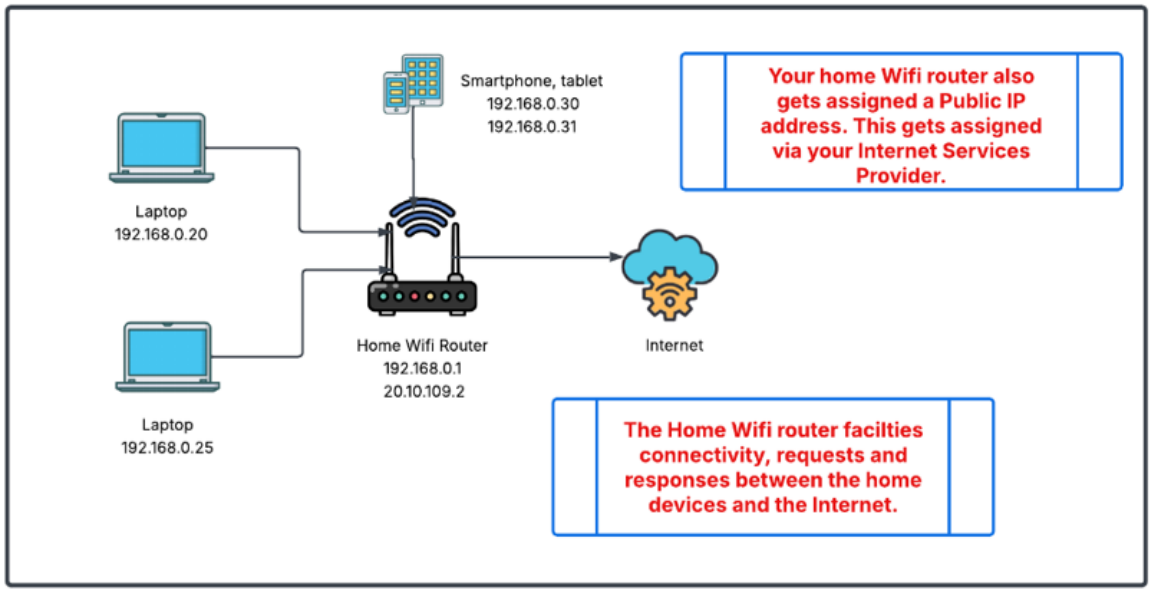
Implement and manage virtual networking

Networking 101 - Networks, devices and IP addresses – Basics

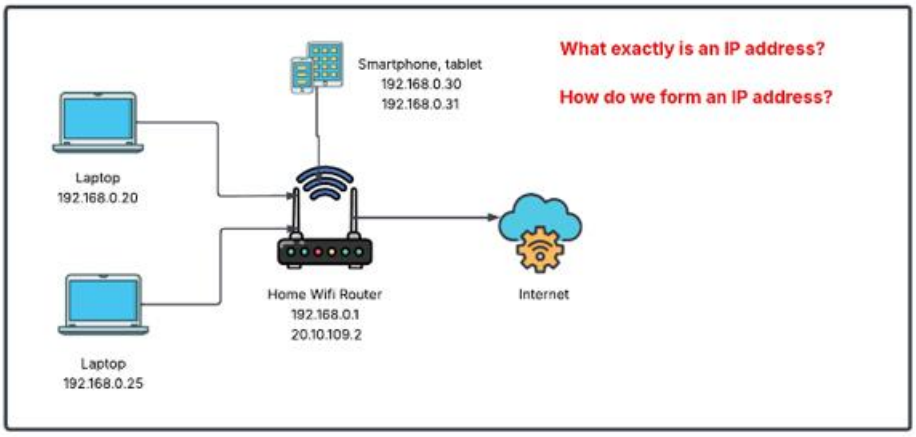





Smartphone, tablet
192.168.0.30
192.168.0.31
Laptop
192.168.0.20
Home Wifi Router
192.168.0.1
Laptop
192.168.0.25
All of the devices on the network also get an IP address. An IP address is used to identify a device on the network
Your devices on the network get a Private IP address. These Private IP addresses are not used to communicate with the Internet
Your Home Wifi router becomes the default gateway. It gets an IP address normally assigned as 192.168.0.1
All traffic from the devices to the Internet gets routed via the Home Wifi Router.

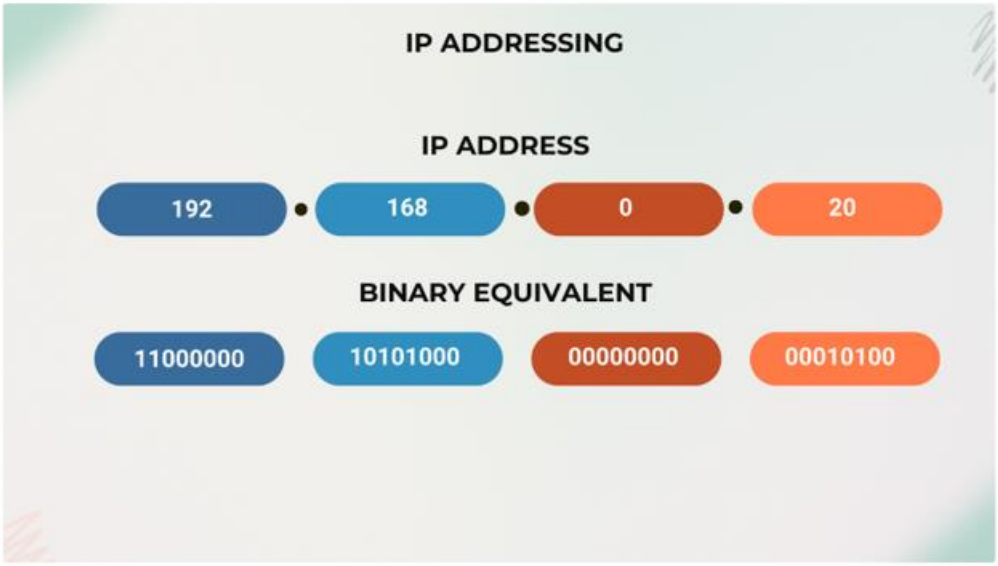

Smartphone, tablet
192.168.0.30
192.168.0.31
Laptop
192.168.0.20
Home Wifi Router
192.168.0.1
20.10.109.2
Internet
Laptop
192.168.0.25
Your home Wifi router also gets assigned a Public IP address. This gets assigned via your Internet Services Provider.
The Home Wifi router facilties connectivity, requests and responses between the home devices and the Internet.

Networking 101 - IP addresses

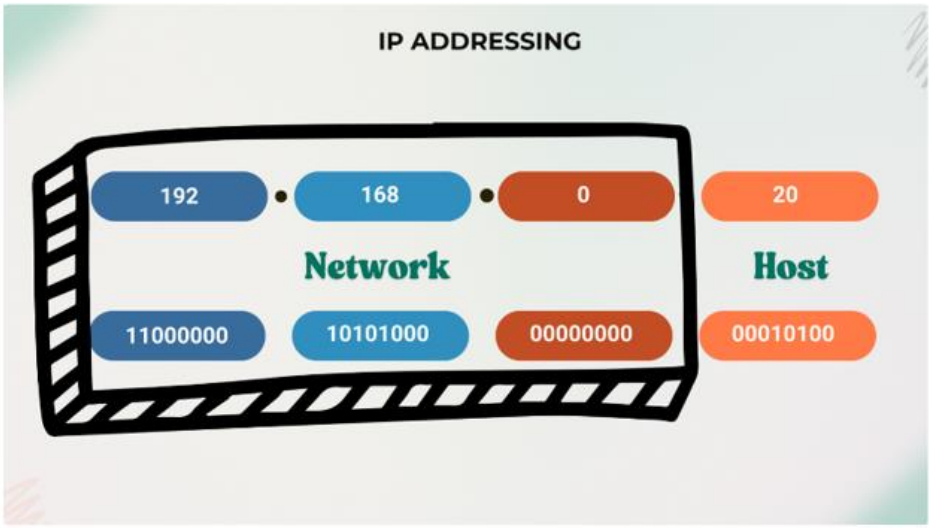


An IP address helps to uniquely identify each device on the network.

An IP address is actually a string of binary numbers. It's actually split into 4 octets , with each octet containing 8 binary bits.

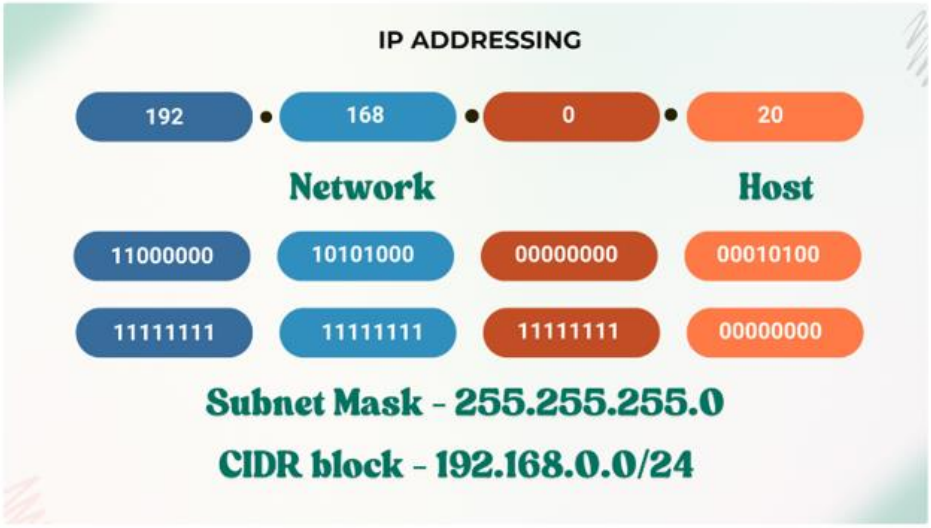


There is a portion of the IP that signifies the network and a portion that signifies the host.



Here as an example, the first 24 bits are used to represent the network. The remaining 8 bits are used to represent the host.

We normally use a subnet mask to segregate what is the network part and what is the host part.



CIDR stands for Classless Inter-Domain Routing. It helps to define a range of IP addresses.

Here 192.168.0.0 specifies the starting IP address of the block.

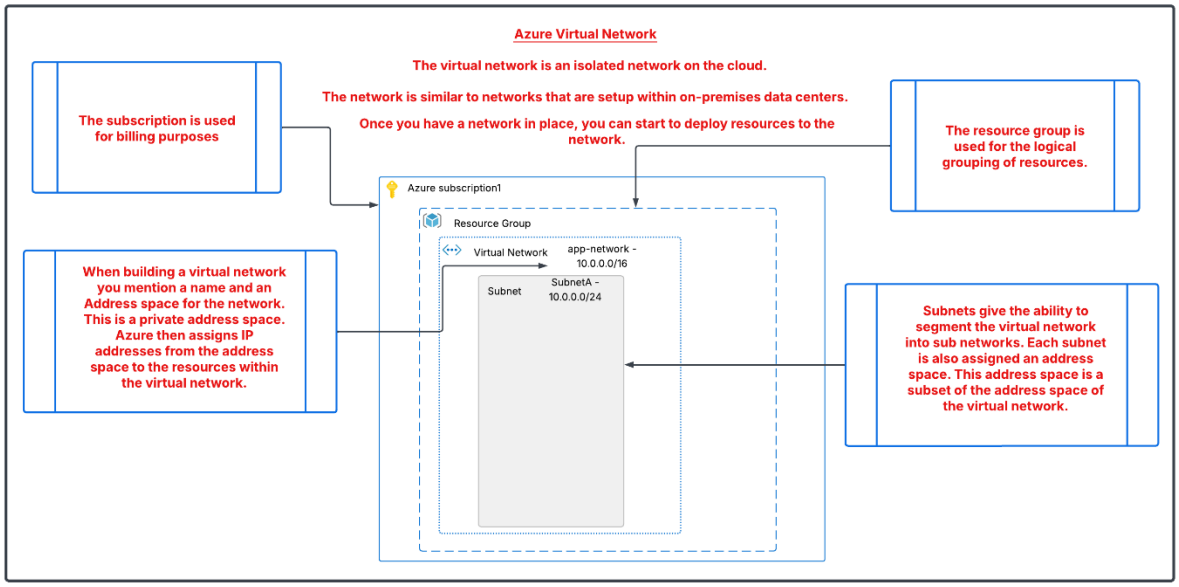
/24 means that the first 24 bits are reserved for the network part and the remaining 8 bits can be used for host addresses.

Introduction to Virtual Networks in Azure

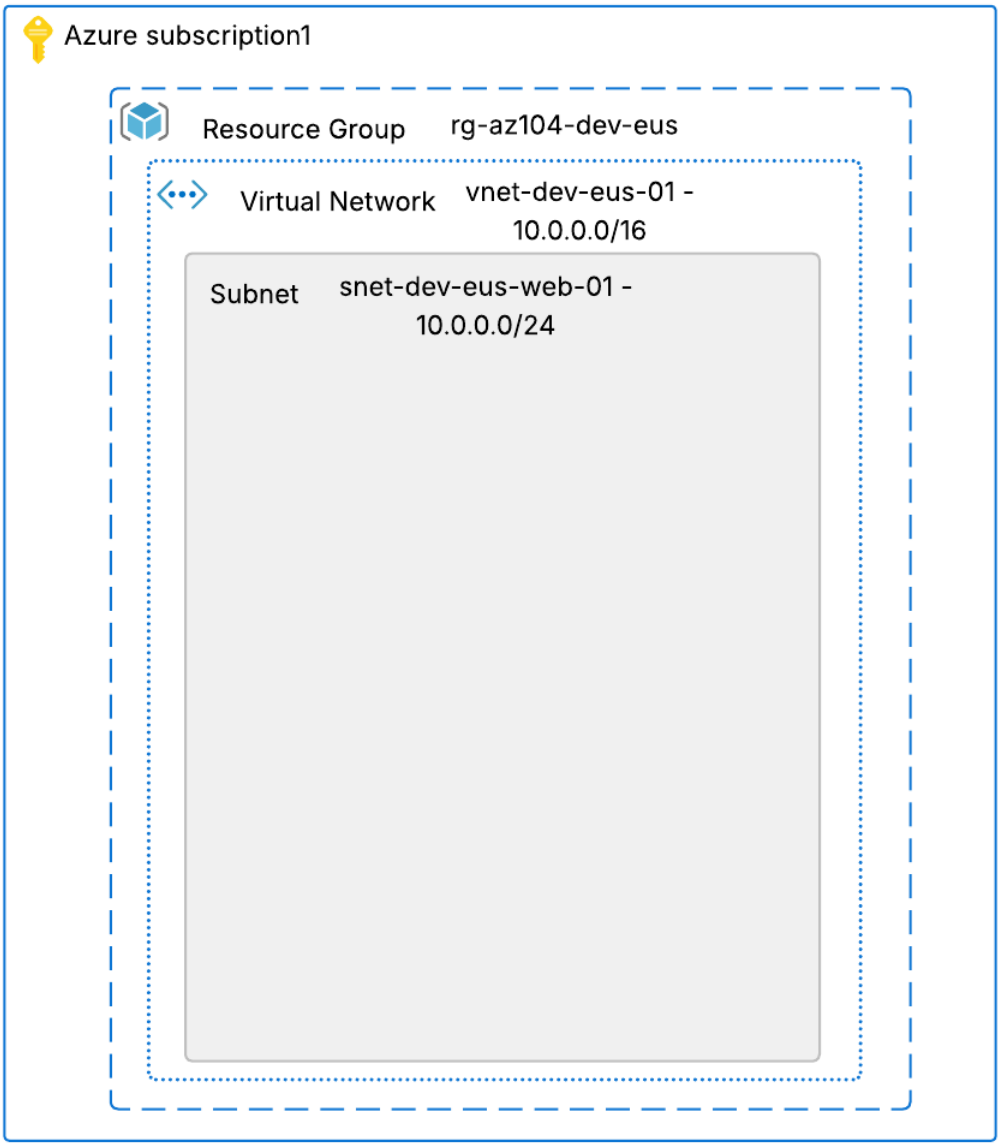
In order to build a physical network we first have to buy and setup hardware routers, switches. Setup access points, setup the underlying cabling.

Things can become messy and complicated pretty fast if you don't streamline the entire setup. When you start creating multiple different networks for different setups, the complications just increase.

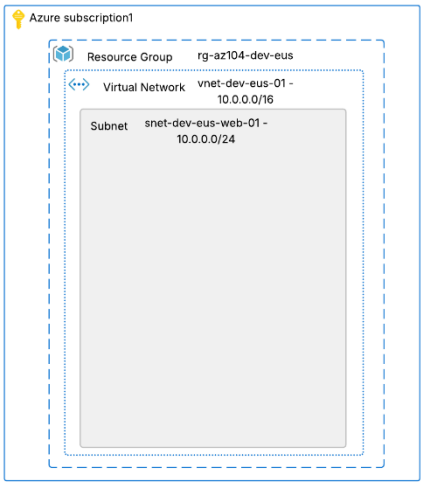
And you need to invest in buying the required hardware to setup the network.



Lab - Building an Azure virtual network



Understanding subnets

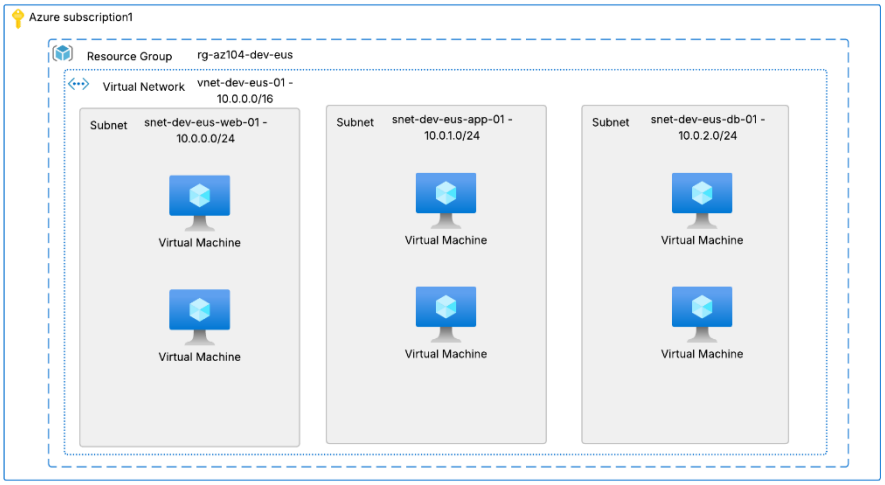


Now let's have an overview onto the concept of subnets.

The virtual networks help to define a private address space in Azure. The subnets help is to slice the network into smaller chunks that can be managed seperately.

Subnets are also defined as a CIDR prefix. This is a subset of the address space of the virtual network.

With the help of subnets, we can segregate workloads, apply different security rules for each subnet.



The above diagram depicts a simple architecture of a virtual network hosting an application.
We have three layers for the application architecture.

1. snet-dev-eus-web-01 could host the Application Gateway machines. This would take in the traffic from the Internet.
2. snet-dev-eus-app-01 could host the virtual machines that host the actual application.
3. snet-dev-eus-db-01 would host the database machines

a) In this architecture, traffic from the Internet would be allowed onto machines in snet-dev-eus-web-01.

b) No traffic from the Internet would be allowed onto machines in either snet-dev-eus-app-01 or snet-dev-eus-db-01.

c) Machines in snet-dev-eus-web-01 would communicate with machines in snet-dev-eus-app-01.

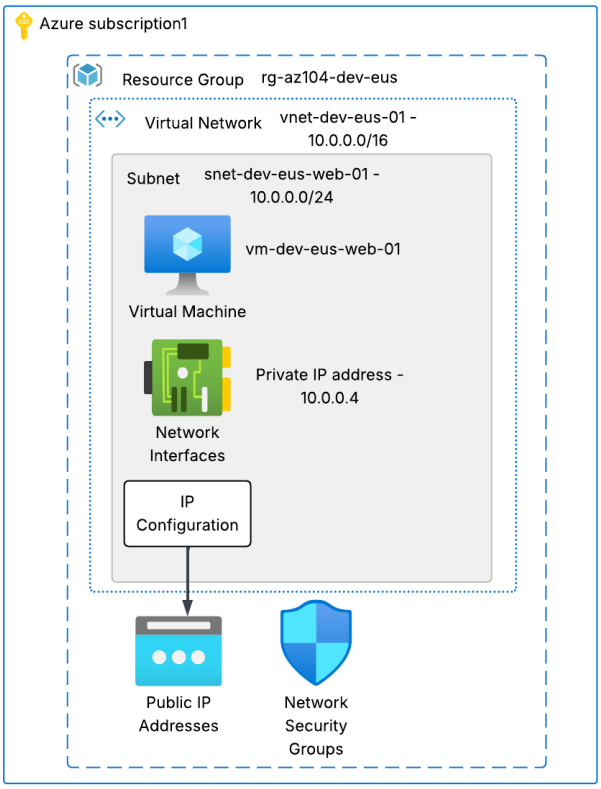
d) Machines in snet-dev-eus-app-01 would communicate with machines in snet-dev-eus-db-01.

Azure Subnets

a) The smallest range you can define for the subnet is /29

b) Azure reserves 5 IP's for each subnet. For example if you have a subnet with the address prefix of 10.0.0.0/24
i) 10.0.0.0 - This is reserved as the network address.
ii) 10.0.0.1 - This is reserved for Azure's default gateway.
iii) 10.0.0.2 and 10.0.0.3 - This is mapped to Azure-provided DNS
iv) 10.0.0.255 - This is the broadcast address

Lab - Deploying an Azure virtual machine to the network



Now let's deploy an Azure virtual machine to the virtual network.

Let's base this on Ubuntu server and for this section remember let's focus on the core networking aspects.

Let's inspect the network interface, the network settings, the IP configuration, the Private IP address, the Public IP address.

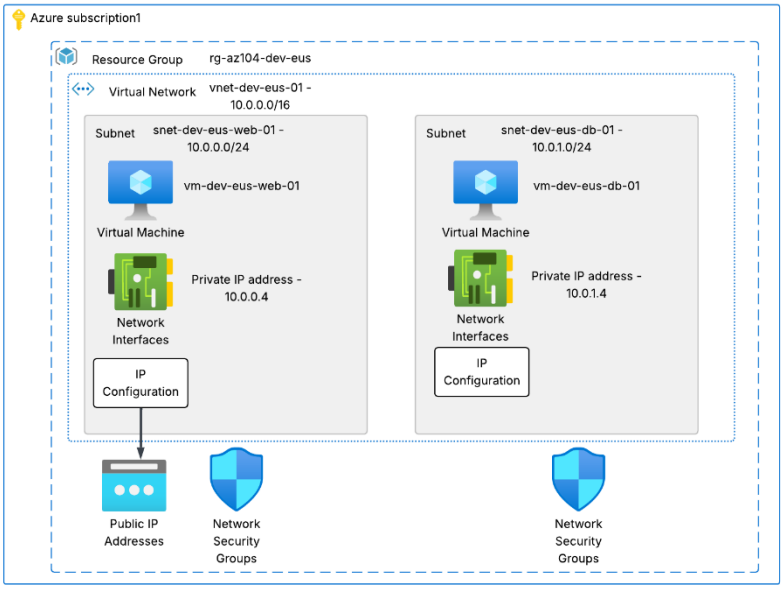
Lab - Let's deploy another machine to another subnet

Now let's build another subnet.

In that subnet let's deploy another virtual machine based on Ubuntu server.

Let's imagine a database server hosted on this machine.

This time around let's not assign a Public IP address to this machine.



Understanding IP addresses

Private IP addresses

These are IP addresses that come from the subnet. These are not routable over the public Internet.

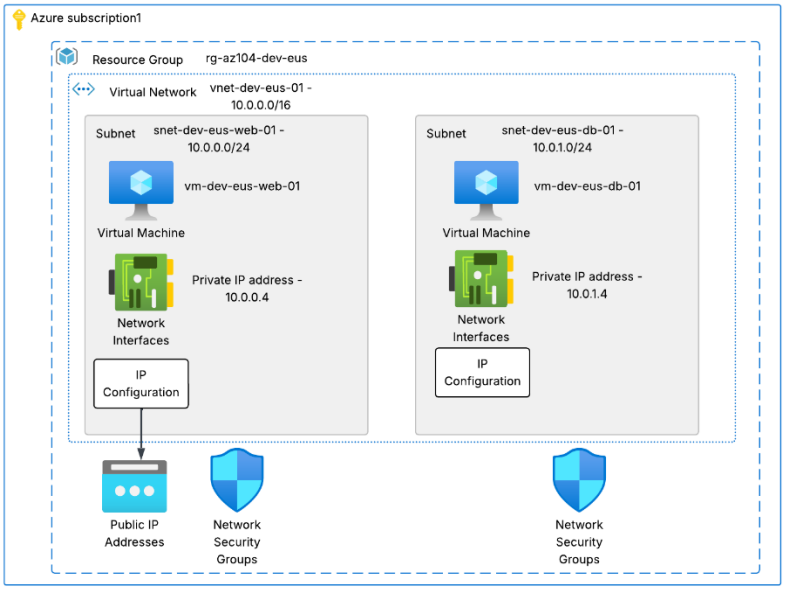
Public IP addresses

These are assigned by Azure. These are routable over the global internet.

With the help of Public IP addresses resources from the Internet can communicate with the virtual machine.

Dynamic Public IP addresses - Here the address is assigned only when the Public IP is assigned to a resource. The address is released when you stop or delete the resource.

Static Public IP address - Here the address is assigned during resource creation. The address is released when the resource is deleted.



Lab - Adding a secondary network interface

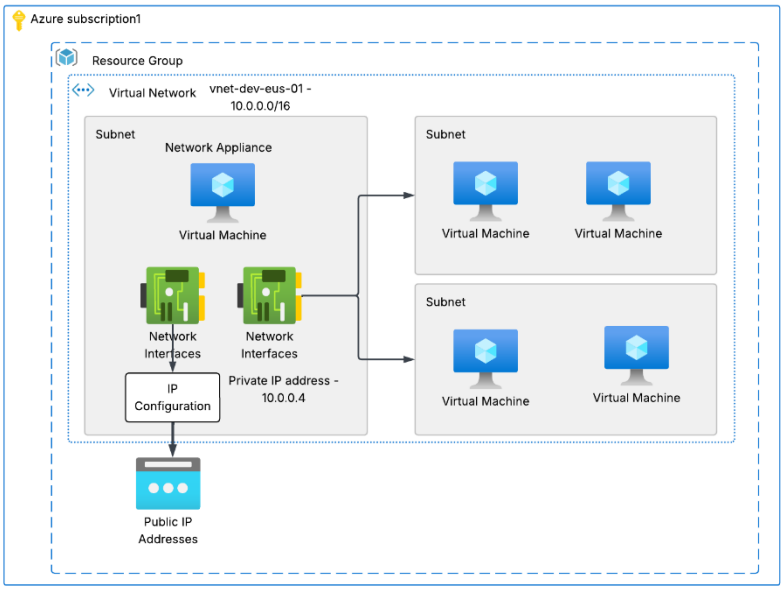
Normally having one network interface is more than enough for a virtual machine.

But there can be cases wherein you might need a secondary network interface.

Let's say you have a machine that also works as a network virtual appliance. Here one Network interface can be used to accept traffic from the Internet.

The other network interface could forward traffic to the internal machines.

Here you have a segregation from a security standpoint with regards to the traffic.



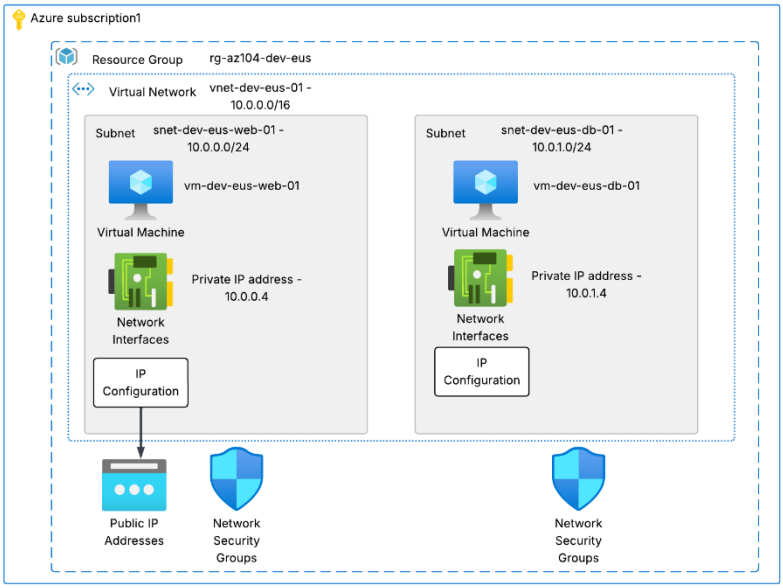
About Network security groups

Network Security Groups

This is a lightweight, stateful packet filter that you can attach to a network interface or a subnet.

Here you add Inbound and Outbound rules that could allow or deny traffic.

These rules are stateful in nature. That means if an inbound request is allowed, the return traffic is automatically allowed.



Rule definition

a) Direction - Whether it's an Inbound or Outbound rule.

b) Priority - The lower priority rules get evaluated first.

c) Action - Whether this rule will Allow or Deny traffic.

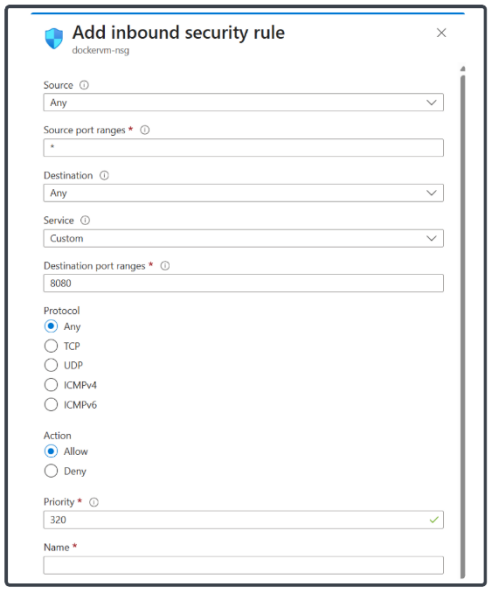
d) What is the Protocol - TCP, UDP, ICMP or Any.

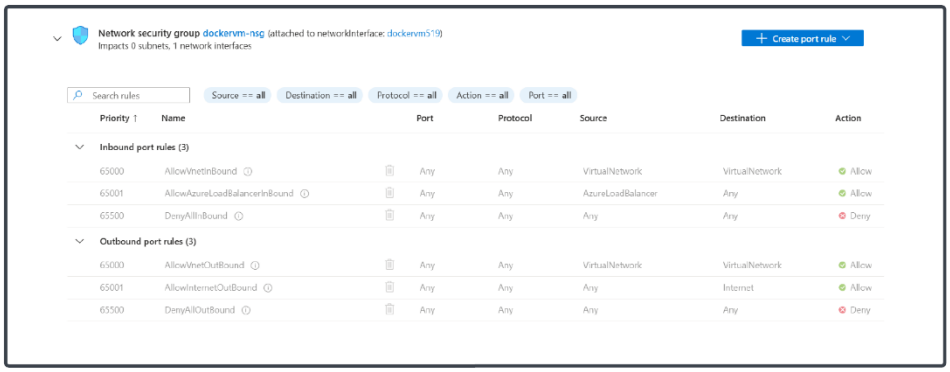
e) The source of the request - This could be a CIDR, A service tag or an Application security group.

f) The source port number

g) Destination - This could be a CIDR, A service tag or an Application security group.

h) The destination port number.





Lab - Network Security Groups - Getting started

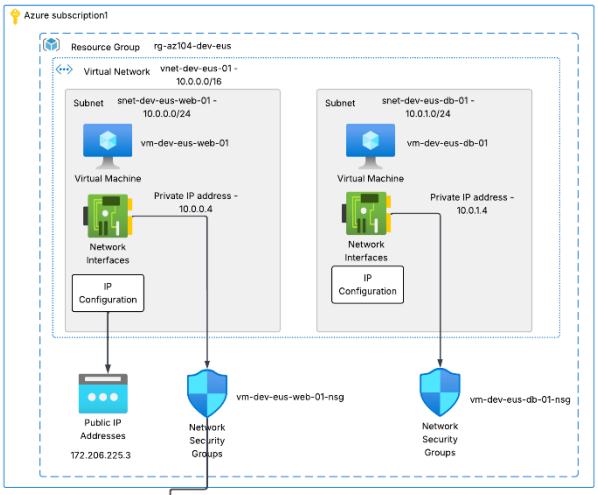
Let's implement the following in this lab

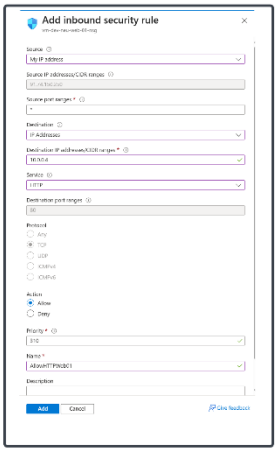
1. Install NGINX on vm-dev-eus-web-01

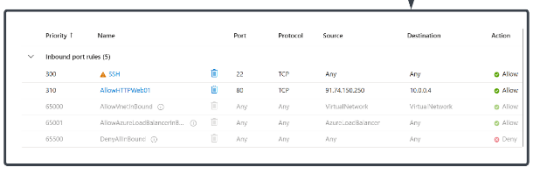
2. Create an Inbound rule to allow traffic from the Internet onto vm-dev-eus-web-01.

3. For the rule, we can be more specific when it comes to the source and destination IP address.

Remember that the Network Security Groups are currently attached to the Network Interface.







Lab - Network Security Groups - Understanding priorities

Now let's understand how priorities work when it comes to Network Security Groups.

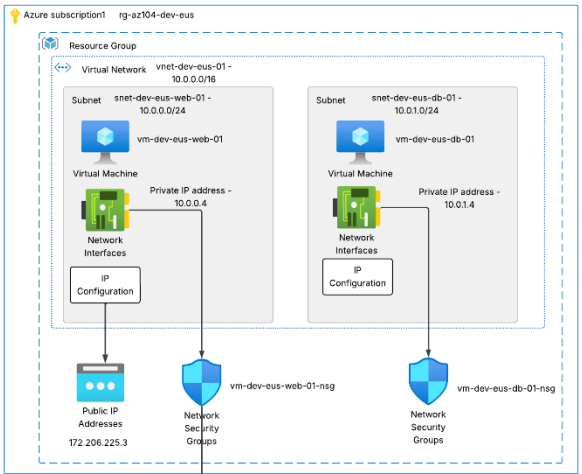
Let's add a rule to deny traffic on a specific port range.

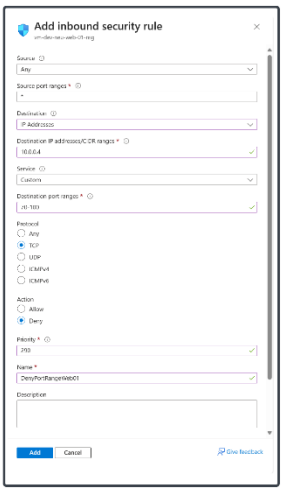
This port range will also include port 80.

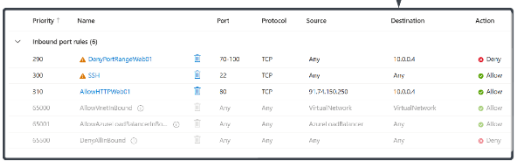
Also place a priority for the rule such that it comes higher up in the list.

1. What is the impact to HTTP requests to the web server

2. Can we still SSH onto the server?







Lab - Network Security Groups - Inter VM communication

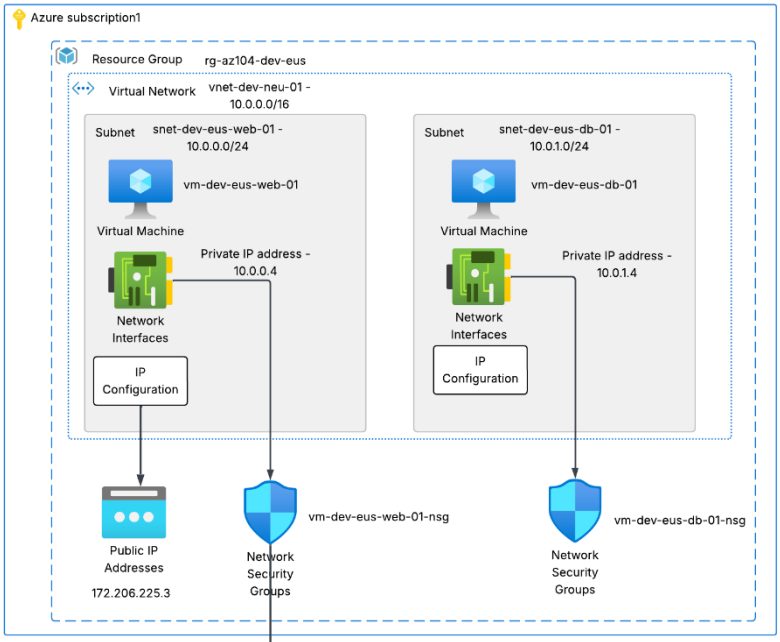
Now let's delete the custom rules we created. Let's also delete the SSH rule as well. Let's have a clean slate when it comes to the rules.

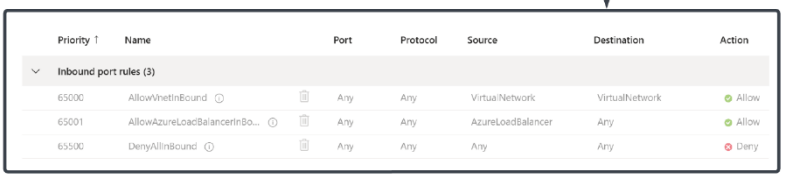
1. Now from vm-dev-eus-db-01 , can we reach the web server running on vm-dev-eus-web-01.

2. Can we ssh from our machine?

3. Can we ssh from vm-dev-eus-db-01 to vm-dev-eus-web-01?

What's the rationale behind our results.



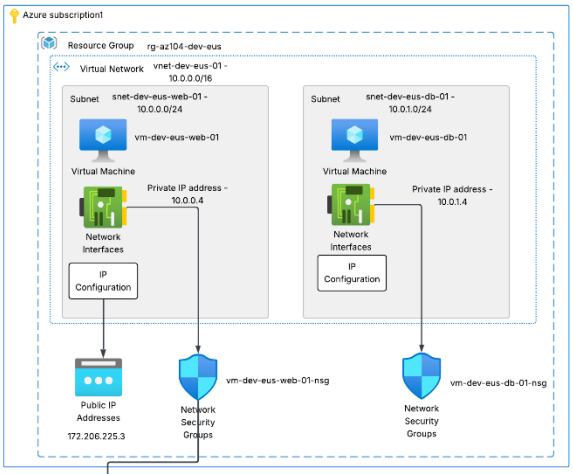


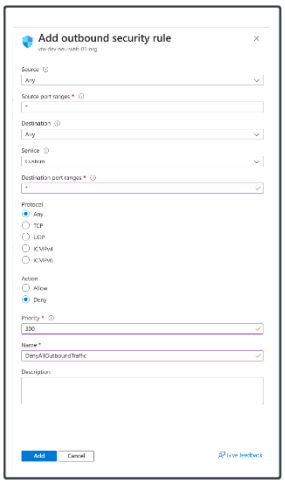
Lab - Network Security Groups - Outbound rules

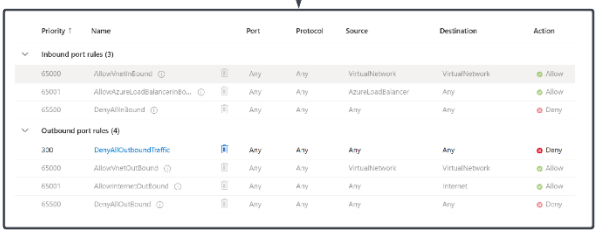
Now let's work a little on Outbound rules.

1. Let's first add a rule to Deny all Outbound traffic. Let's add this rule to vm-dev-eus-web-01-nsg. Now can we still reach the web server running on vm-dev-eus-web-01 from vm-dev-eus-db-01.

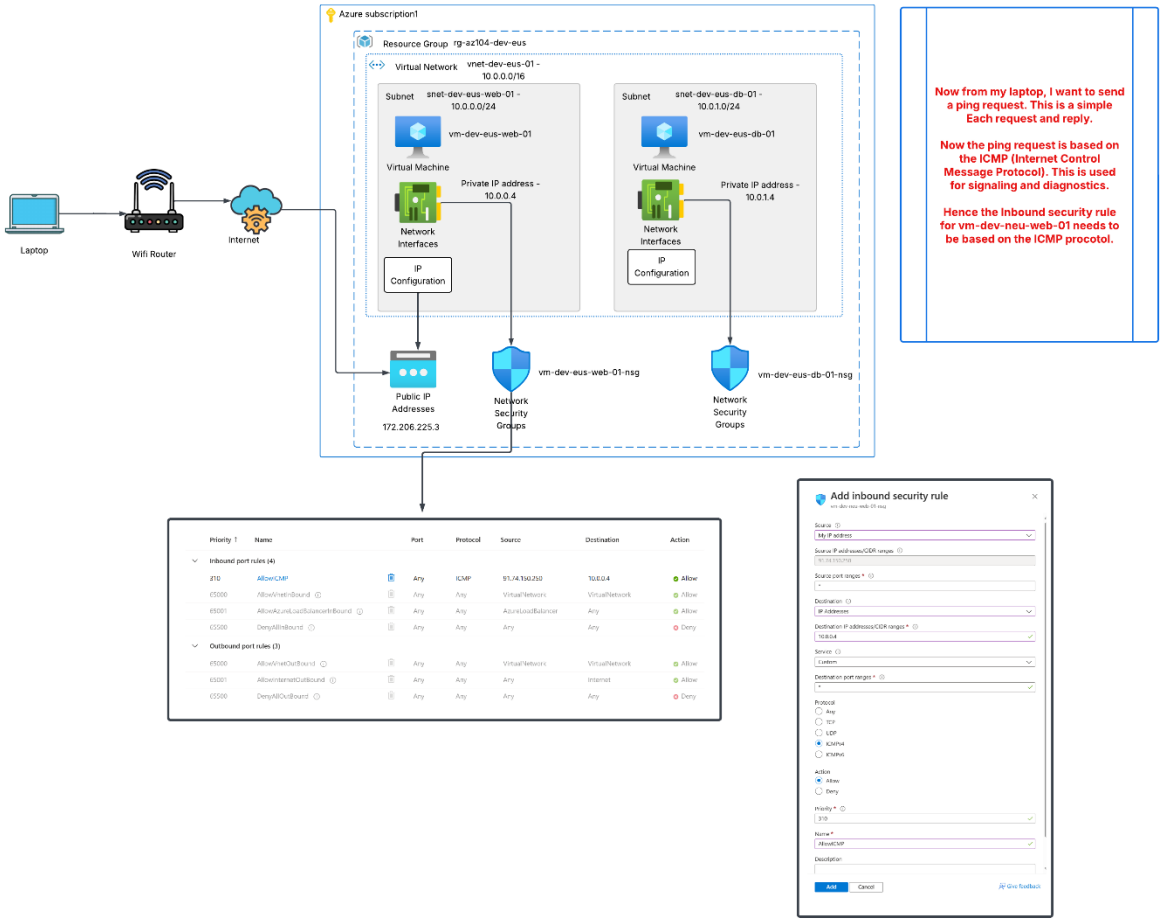
Here the assumption would be No - because the Deny All Outbound would not allow the response to flow outward from vm-dev-eus-web-01 to vm-dev-eus-db-01.



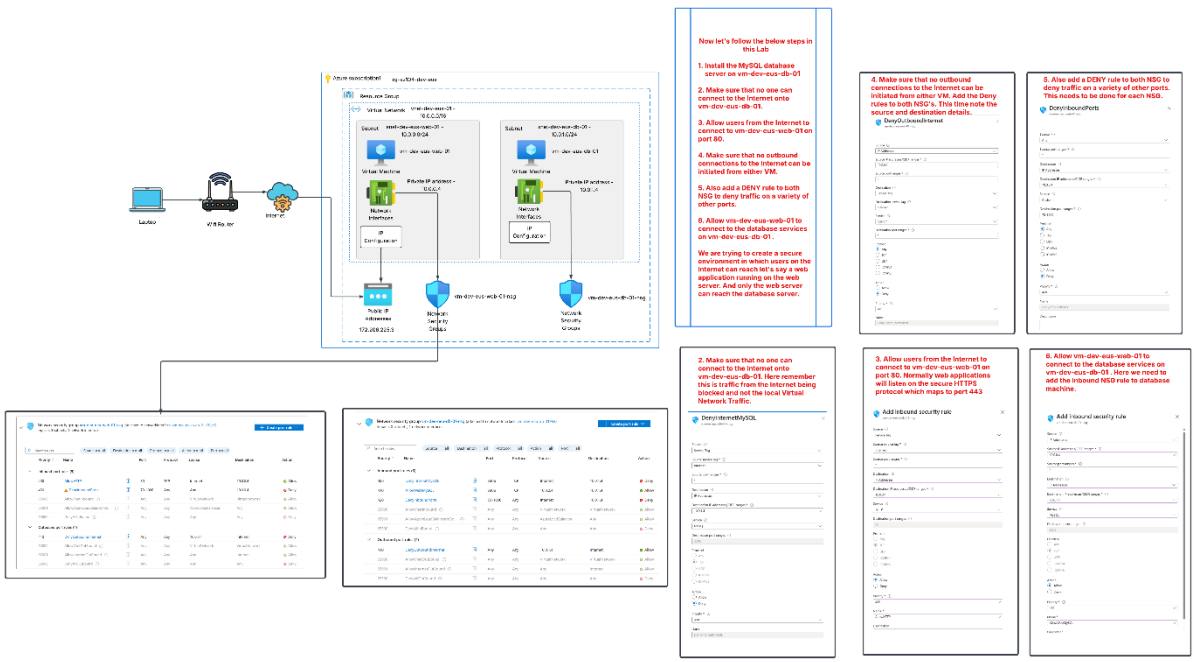




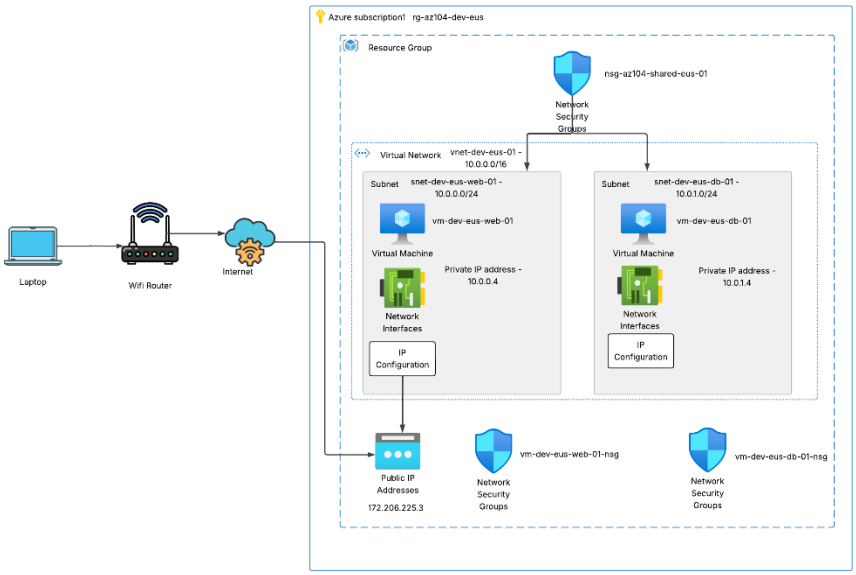
Lab - Network Security Groups - ICMP Protocol



Lab - Network Security Groups - Two-tier architecture



Lab - Network Security Groups – Subnets



When it comes to Network Security Groups, we can attach an NSG to a subnet and a network interface.

Let's implement the following steps in this lab

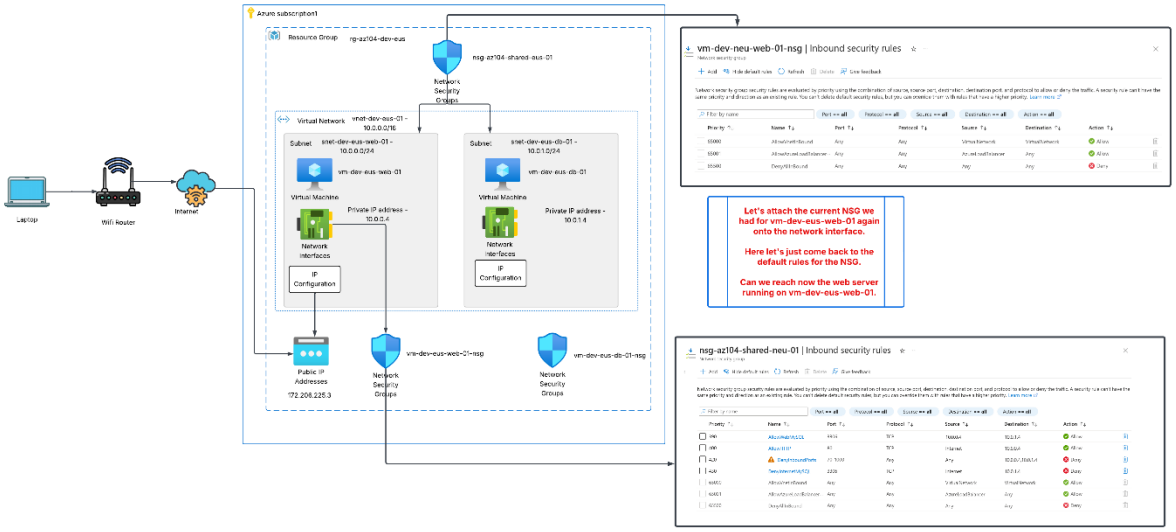
1. Create a new Network Security Group.

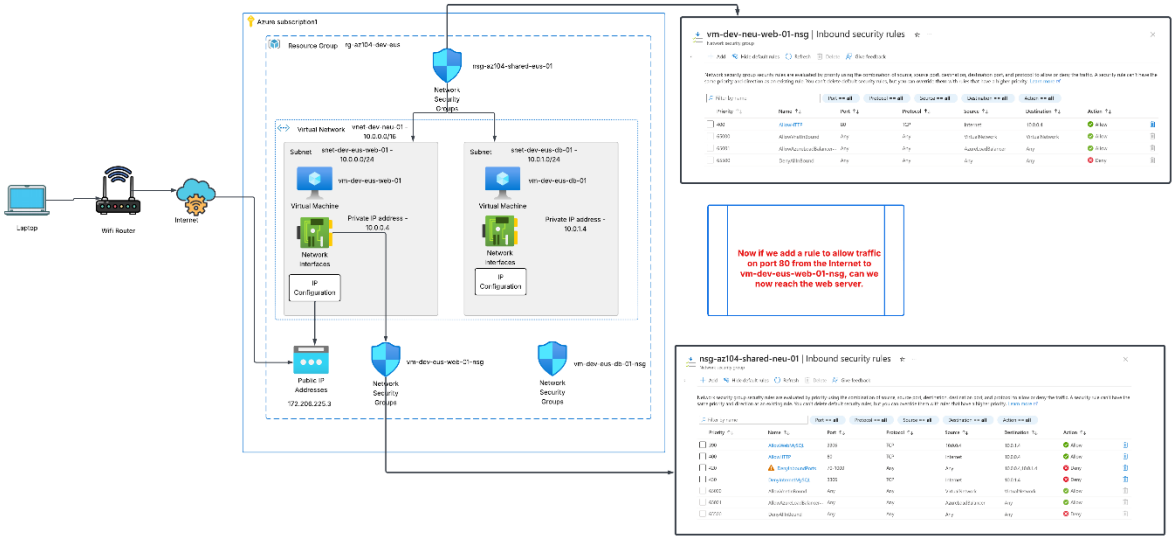
2. Detach the current Network Security Groups which are at the network interface layer.

3. Attach the new Network Security Group to the subnets holding the machines.

4. Define the required rules.

Lab - Network Security Group - Subnet and Network Interface

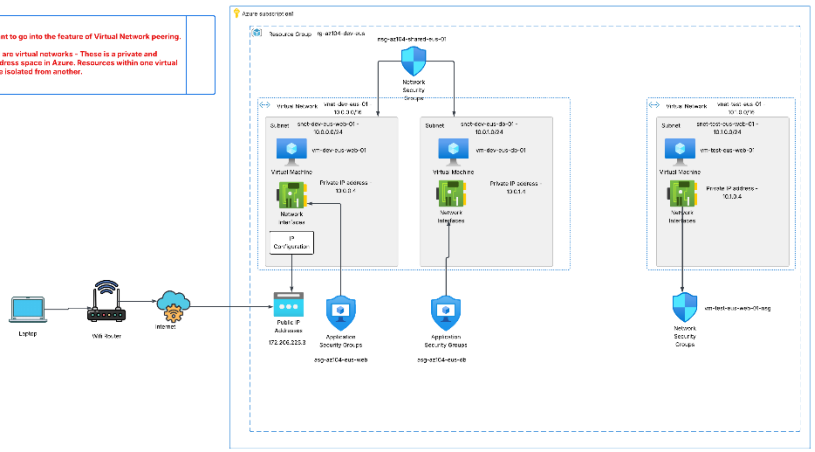




Lab - Virtual Network Peering

Now we want to go into the feature of Virtual Network peering.

Again what are virtual networks - These is a private and isolated address space in Azure. Resources within one virtual network are isolated from another.



In order to understand virtual network peering, let's perform the following steps

1. Create a new virtual machine in a new virtual network in the same region. Let's imagine we are trying to replicate our infrastructure onto a test environment.

Let's again base the machine on Ubuntu server - Note that in environments you can have a mix of both Windows and Linux-based machines. Let's not assign a Public IP to this machine.

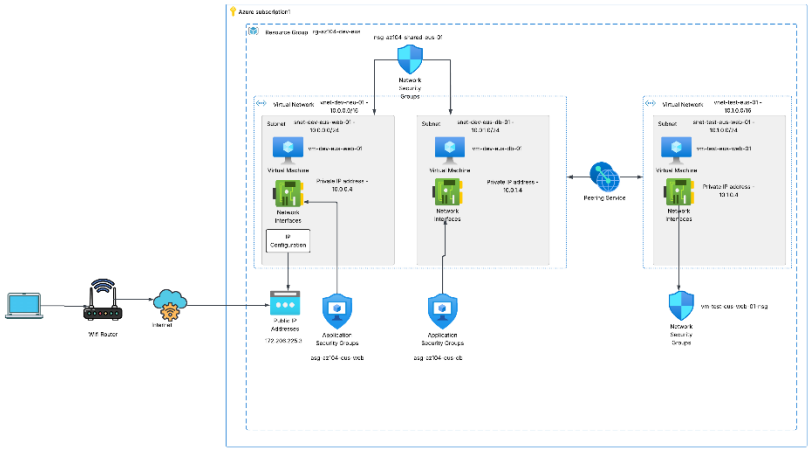
Also let's only have one NSG for our dev setup, where we use the shared NSG attached to the subnets. Also for the shared NSG, let's keep things a little simpler by not having any Deny rules.

Couple of side queries - Can you associate nsg-az104-shared-eus-01 with snet-test-eus-web-01?

Can you label the NIC attached to vm-test-eus-web-01 with either Application Security Group?

But the main query we want to tackle is, from vm-test-eus-web-01 can you reach the web server running on vm-dev-eus-web-01?

Not possible - This is because both virtual networks are isolated from each other. But we can connect the networks together via virtual network peering.



Azure Bastion

Azure Bastion

This is a fully managed Platform as a service that allows you to securely connect to your virual machines via private IP addresses. Here you can RDP into the Windows machines and SSH into the Linux-based machines.

The benefit here is that you don't need to allocate Public IP addresses to the machines for connectivity. It reduces the surface area attack for your VM's. Here you can just use the Bastion to connect to the machines.

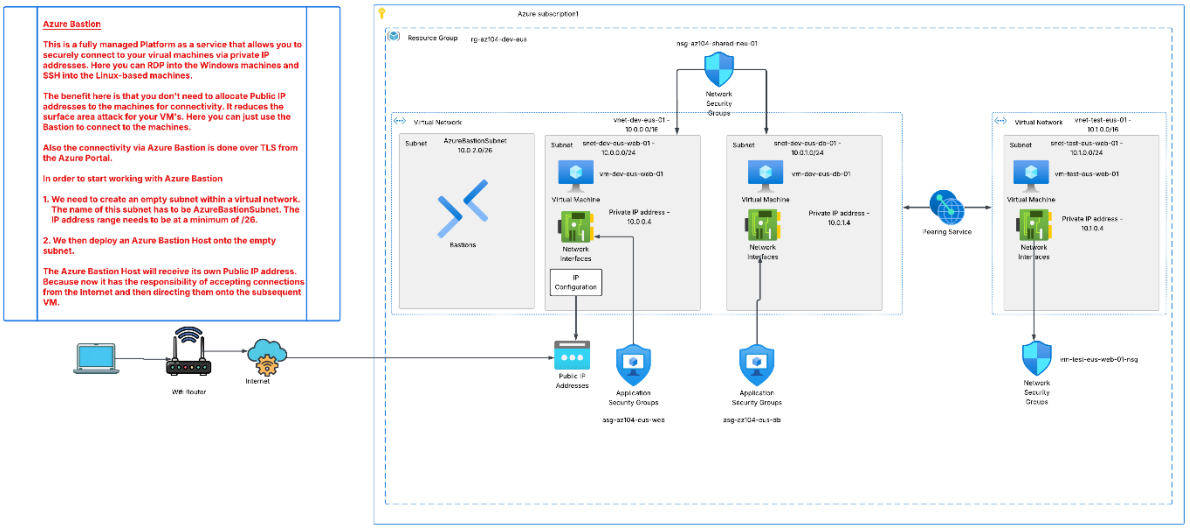
Also the connectivity via Azure Bastion is done over TLS from the Azure Portal.

In order to start working with Azure Bastion

1. We need to create an empty subnet within a virtual network. The name of this subnet has to be AzureBastionSubnet. The IP address range needs to be at a minimum of /26.

2. We then deploy an Azure Bastion Host onto the empty subnet.

The Azure Bastion Host will receive its own Public IP address. Because now it has the responsibility of accepting connections from the Internet and then directing them onto the subsequent VM.



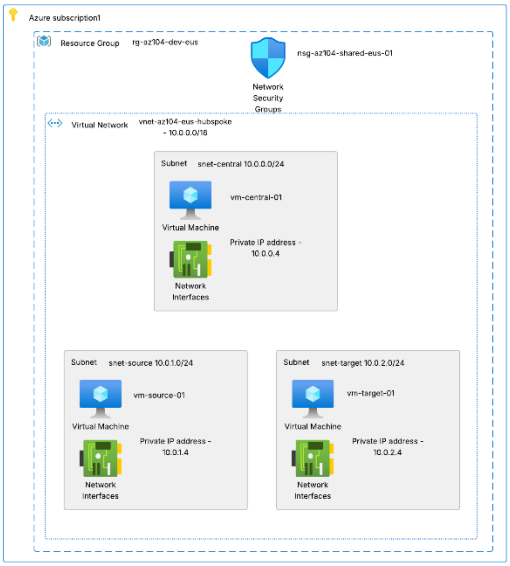
User defined routes

1. First let's create our base infrastructure - No Public IP addresses. Just one centralized network security group.

2. Let's inspect the system defined routes.

3. Then install nginx on vm-target-01.

4. Confirm that we can reach this web server from vm-source-01.



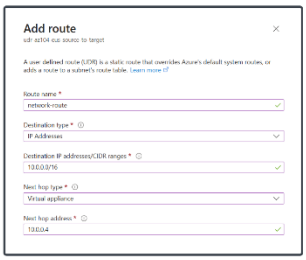
Now we are going to route all traffic from subnets snet-source and snet-target to the go via the vm-central-01 vm.

1. We will first create a route table.

2. Then we add a route that specifies that any traffic for the virtual network needs to flow via the central vm machine.

3. Then we associate the route table with the snet-source and snet-target subnets.

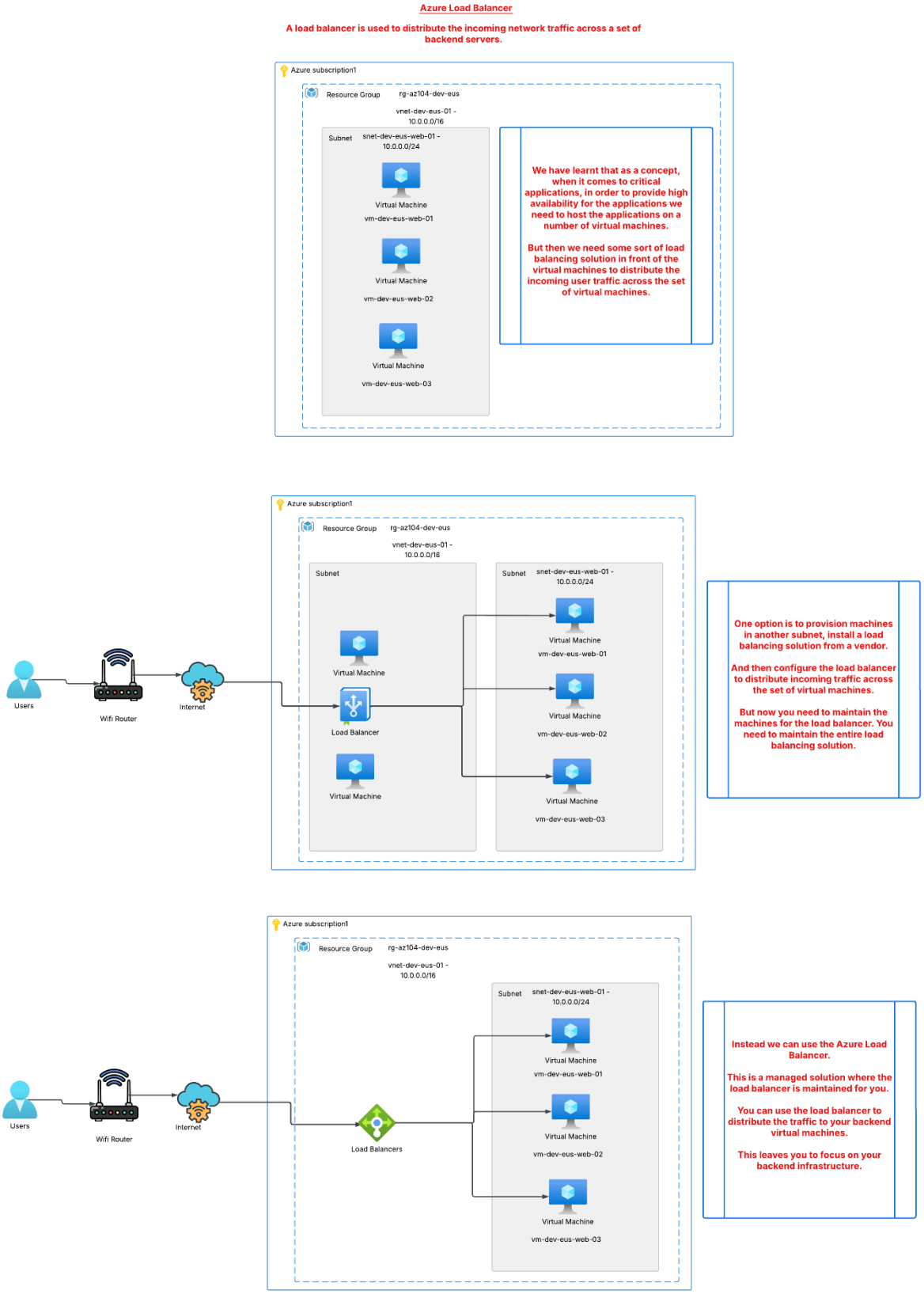
Now test to see if you can reach the web server running on vm-target-01 from vm-source-01



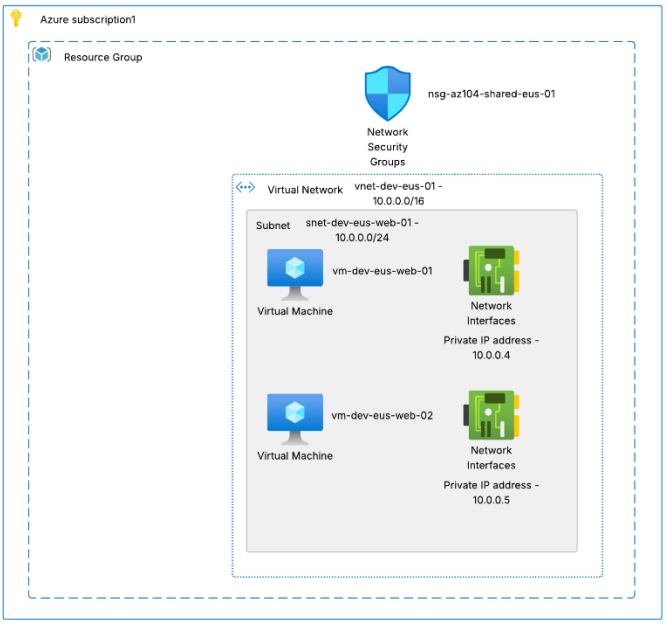
Yes, we now have requests flowing via vm-central-01. But now we need to make sure that this vm can forward packets from the source on its destination.

For this we need to enable IP forwarding at the network interface level and the OS level.

The Azure Load Balancer Service



Lab - Azure Load Balancer - Standard SKU

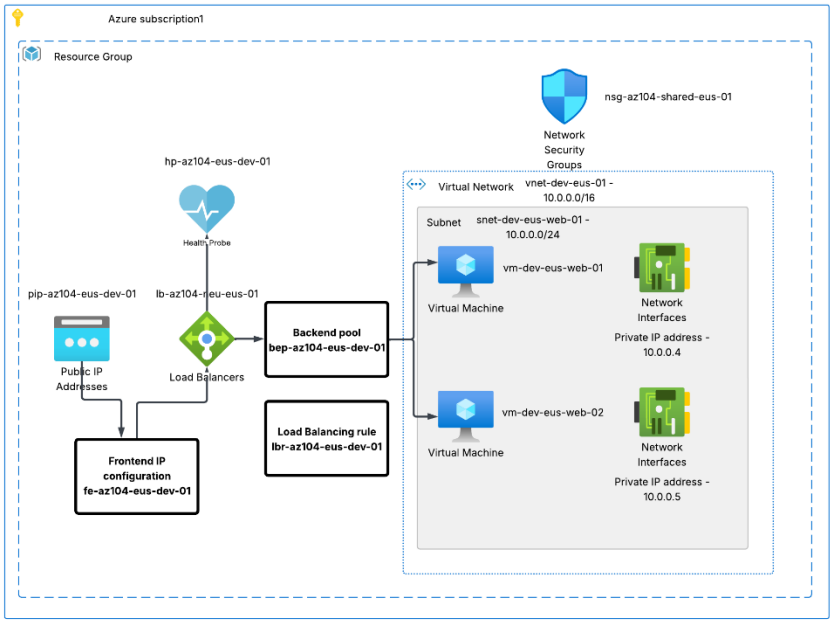


For our setup, we will first create the backend machines that will sit behind the Load Balancer.

The machines will be based on Ubuntu server and let's install NGINX on each server.

We will have one central Network Security Group.

Also we are not going to assign any Public IP addresses to the individual machines.



Now let's deploy the Azure Load Balancer as the next step. This will be configured as

SKU : Standard
Type : Public
Tier: Regional

The Standard Load Balancer , if Public needs to have a Public IP address. The Public IP address becomes part of a frontend-ip configuration. The Public IP address needs to be of the Standard SKU.

We wil configure the remaining aspects after the deployment of the load balancer.

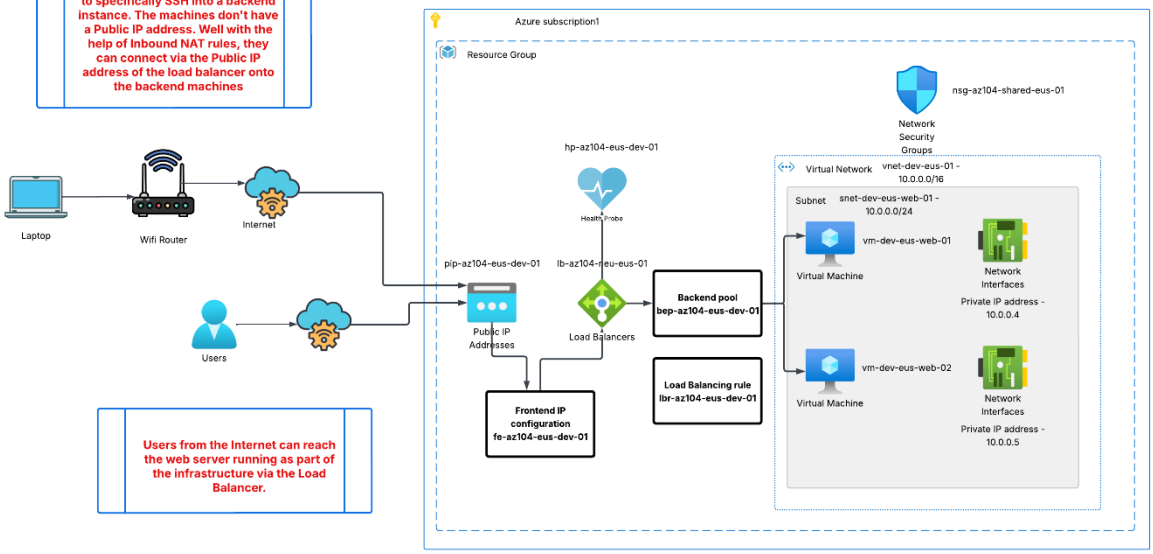
1. Backend pool - After the load balancer is deployed, we configure the backend pool of virtual machines. The Load balancer can the direct user requests onto the machines in the backend pool.

2. Health Probe - This is used to determine if the backend machines are healthy or not.

3. Load Balancing rule - This will instruct the Load Balancer on how to route requests to the backend pool.

Lab - Azure Load Balancer - Standard SKU - Inbound NAT Rules

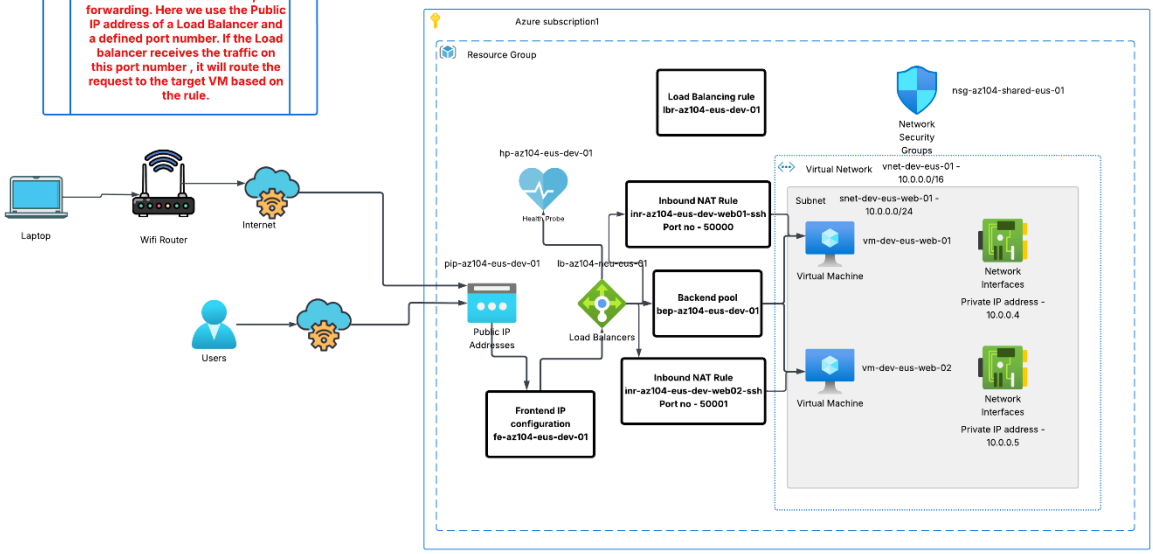
Well what if an administrator wants to specifically SSH into a backend instance. The machines don't have a Public IP address. Well with the help of Inbound NAT rules, they can connect via the Public IP address of the load balancer onto the backend machines



Users from the Internet can reach the web server running as part of the infrastructure via the Load Balancer.

For this we make use of Inbound NAT rules

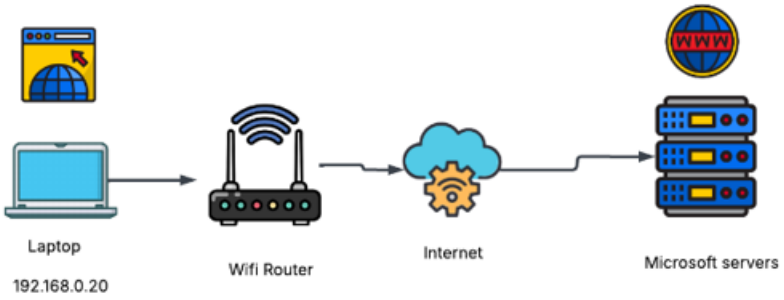
The NAT rules make use of port forwarding. Here we use the Public IP address of a Load Balancer and a defined port number. If the Load balancer receives the traffic on this port number , it will route the request to the target VM based on the rule.



DNS Fundamentals for Azure Administrators

Now when you type a site in your browser, like https://www.microsoft.com, your laptop does not connect to the name , but it needs to establish a connection to the underlying IP address.

In the end for your browser to get the web page, a TCP connection needs to be established between the laptop and the servers hosting the Microsoft site.



So ideally in the browser, we should be typing the IP address of the remote Microsoft servers to get the web application data.

But we as humans are more familiar with remembering names rather than IP addresses. Hence we type in a site name which also is the domain name.

So where does DNS or the Domain Name system fit into all of this. Well how does your machine get the IP address to connect to that actually maps to https://www.microsoft.com

Well the DNS system is like a distributed phone book that helps to turn your domain names into IP addresses.

In the end the DNS system just hosts a set of records. The record just has a mapping of the domain name to the IP address.

Your laptop needs to fetch the IP address mapped to the domain name from the domain name system.

Simple understanding of the steps involved in browsing a site.

1. You type in https://www.microsoft.com in the browser on your laptop.

Laptop

192.168.0.20

2. The browser would first formulate a query to resolve the domain name of www.microsoft.com from the Internet.

3. The query would go to the ISP to see if it can be resolved there.

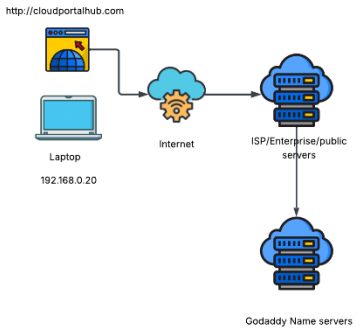
4. If not, the ISP would forward the request to other domain name servers across the Internet.

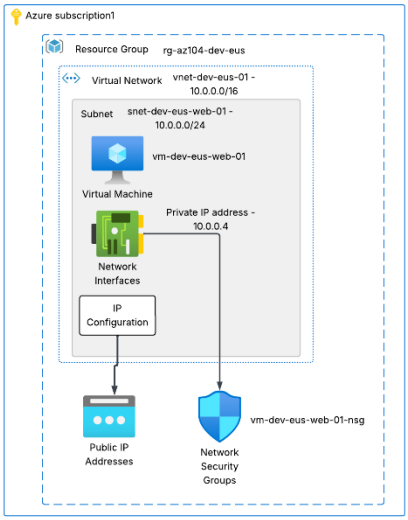
5. Finally the request would reach an authoritative Name server. A name server is a DNS server that answers DNS questions for the domain.

6. The IP address for the domain is sent back to the browser/laptop.

7. Your laptop then makes a connection across the Internet to the Microsoft servers via IP addresses.

An example of setting the DNS name for an Azure virtual machine





In an external domain provider, for a domain we own, we can set DNS records. We can set a simple record that would map the domain name to the IP address.

Now when our browser fires a query to resolve cloudhublearning.com, the request would reach the name servers at Godaddy. The name servers would then return the IP address to the laptop. And then the connection is established between the Public IP address of the VM and my laptop.

The different types of DNS records

A (Address) record - This maps a domain name to an IPv4 address.

CNAME record - This maps one domain name onto another.

NS record - This delegates the responsibility of the DNS zone to the specific name servers.

SOA record This is a Start of Authority record - It has metadata on the zone.

MX record - Mail Exchanger record - Defines which mail servers handle email for the domain.

Designing a Private DNS Foundation in Azure VNets

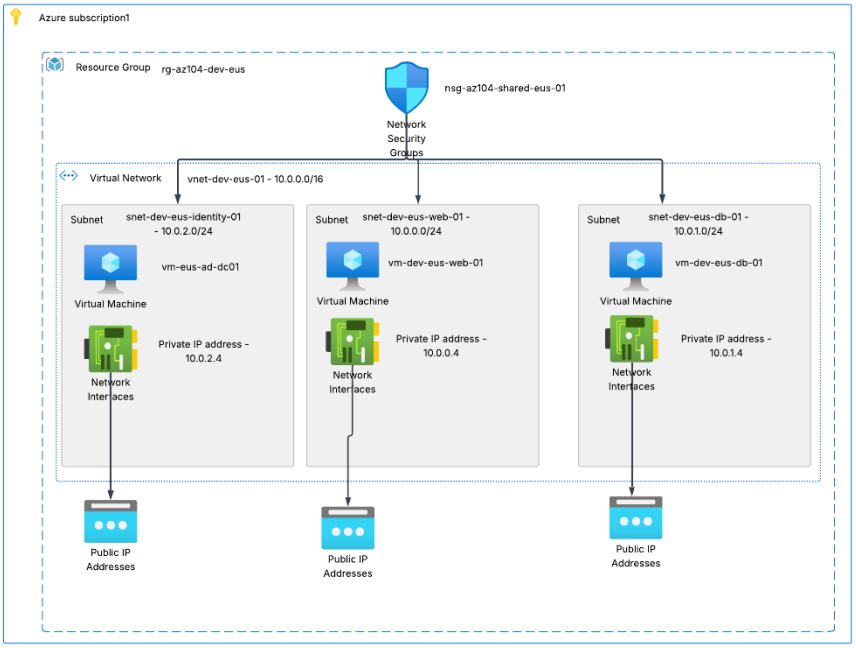
Let's build the environment

1. vm-eus-ad-dc01 will be used to host our local DNS server.

2. We will just have a Public IP for this server so that we can log onto the server and install the required tools.

3. One central Network Security Group.

4. The remaining servers will be Windows as well. On vm-dev-eus-web-01 , let's install Internet Information services and have a sample html file in place.



Lab - Building the Domain Installing AD DS & DNS on Windows Server 2025

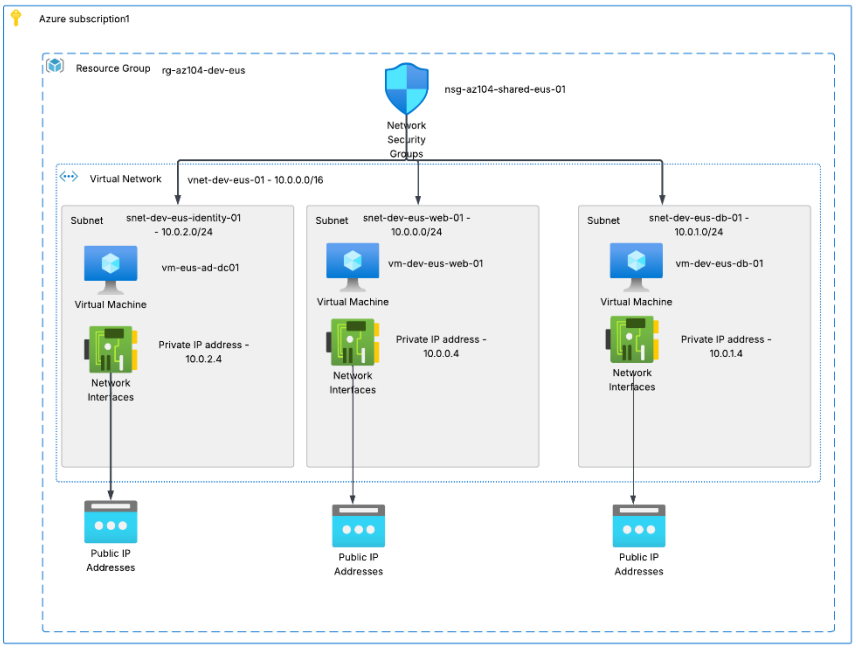
Next steps

1. Make sure the NSG allows RDP connections from the Internet onto the private IP address of vm-eus-ad-dc01.

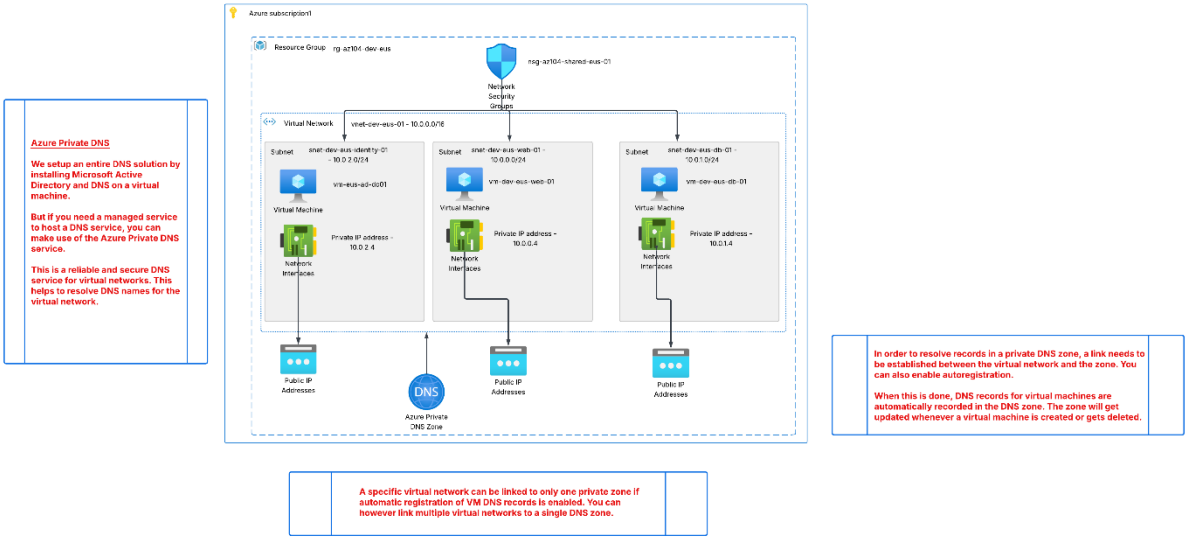
2. Install the Active Directory role onto the machine.

3. Once done, we need to modify the virtual network setting to recognize vm-eus-ad-dc01 as the new DNS server for the virtual network. And then restart the other machines.

4. For each machine - vm-dev-eus-web-01 and vm-dev-eus-db-01, we can join them to the new internal, private domain.



Azure Private DNS



Azure App Service - Virtual network integration

Now in the section of compute we know that there are several ways of hosting your application.

As an example, let's say that you have a web application that needs to connect to a database server.

We could host the web application and database server on Azure virtual machines.

Here the web server machine can have a Public IP address so that users from the Internet can connect to the web server. And then the web server can connect to the database server via private IP addresses.

For such a requirement, we can use the virtual network integration feature of the Azure App service. It allows access from the web app to resources in the virtual network.

But the other way is not possible. No resources from the virtual network can initiate communication with the Azure Web App.

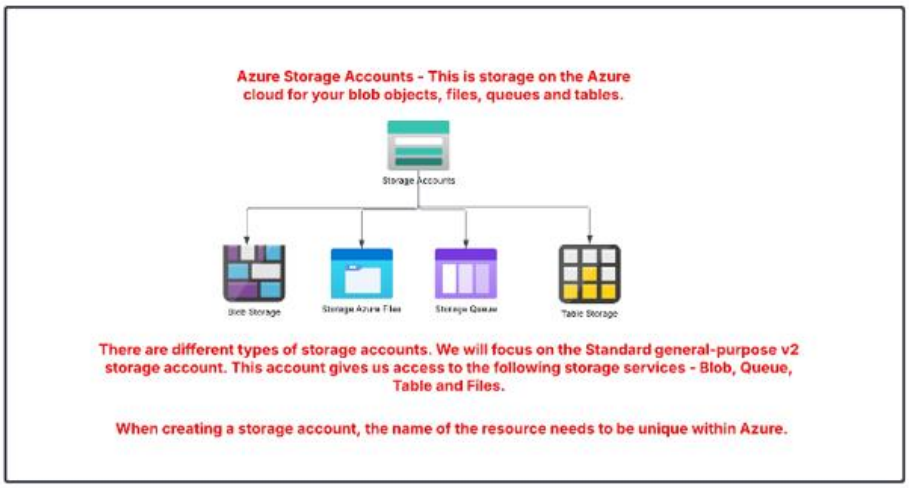
This feature requires the Basic App Service plan or higher.

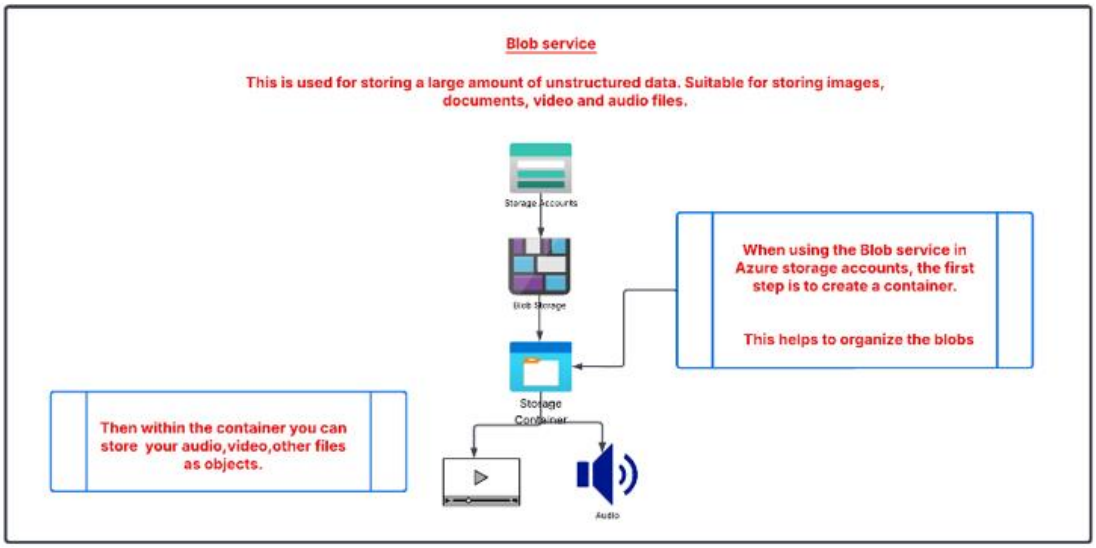
This feature also requires the virtual network to be in the same region as the App service.

To enable virtual network integration, we need to have an empty dedicated subnet in the virtual network. This subnet will use by the virtual network integration feature.

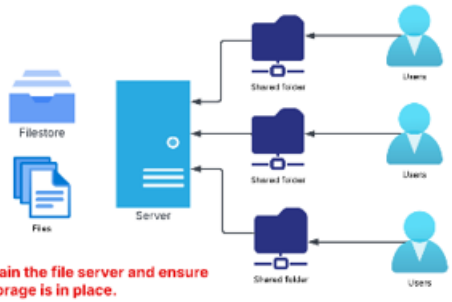
Implement and manage storage

Introduction to Azure Storage Accounts





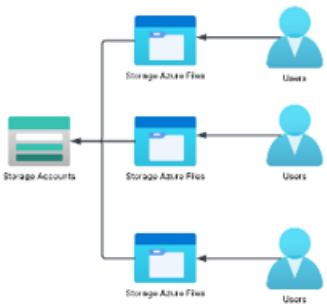
Azure Files



Let's say that in a company, users need access to their own respective file shares to store files.

Here you need to maintain the file server and ensure enough storage is in place.

You can instead opt to use the File service in Azure Storage Accounts. This is a fully managed file share that is accessible via the Server Message Block (SMB) and Network File System (NFS) protocol.



Here the physical storage is managed for you.

If messages need to be shared across multiple application components. Here you need to have the message software and maintain it.

Instead we can make use of the Queue service which provides the basic messaging service.

If an application needs to store data (non-relational structured data), like let's say data about users.

Azure Storage - Blob service - Different authorization techniques

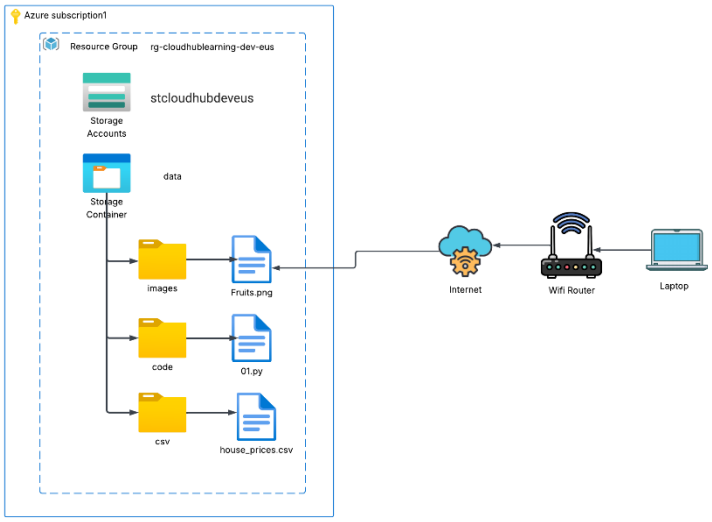
Now there are different ways to authorize users to access data in an Azure storage account. For the moment we will focus on the Blob service.

1. Anonymous access - This can be used if you have a container that hosts images or a static website for which you want to give anonymous access. This is normally disabled by default because its not the most secure way to access content.

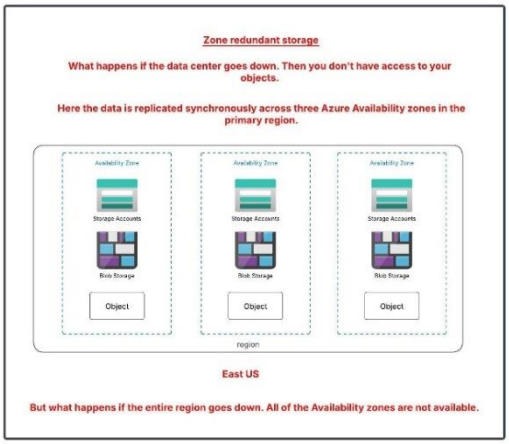
2. Key authorization - Each storage accounts gets two keys. These are like passwords, they are long strings. This gives access to all data in the storage account. It's like the master password for the storage account.

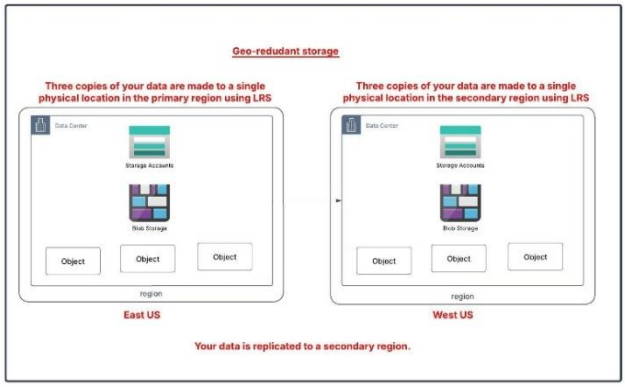
3. Shared Access Signatures - Here you can generate temporary, fine-grained tokens. These SAS tokens can limit the resource for access, what actions can be performed etc. This is ideal for giving short-term access to data.

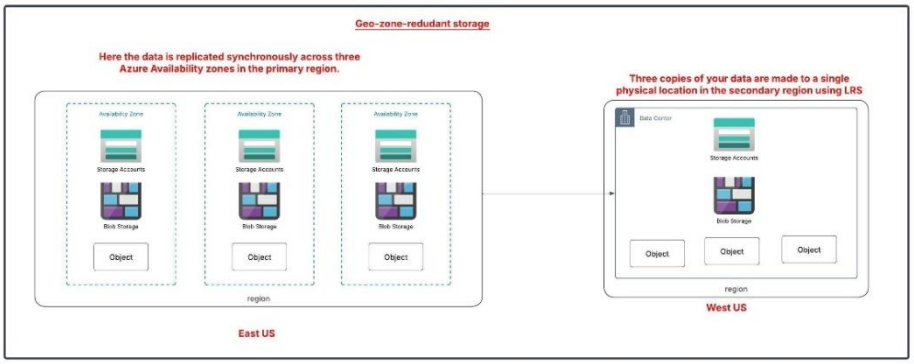
4. Microsoft Entra ID - Here you assign Role-based access control to your user identities.



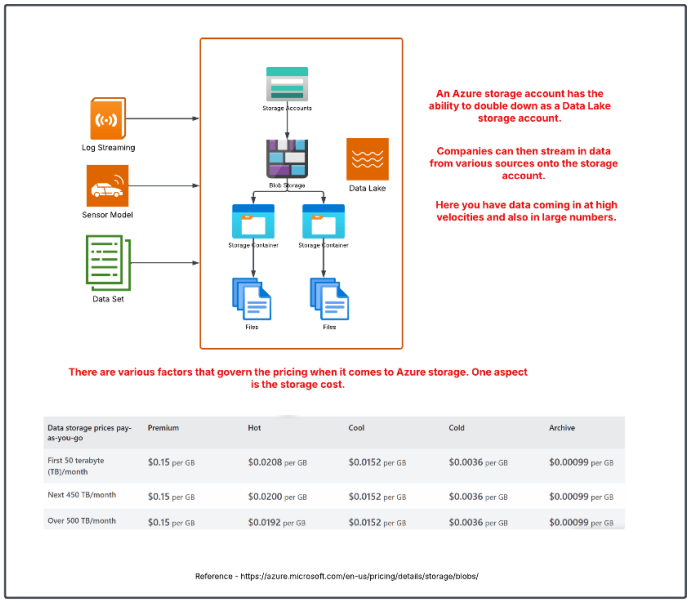
Azure Storage Accounts - Data Redundancy







Storage Accounts - Introduction to Access Tiers



An Azure storage account has the ability to double down as a Data Lake storage account.

Companies can then stream in data from various sources onto the storage account.

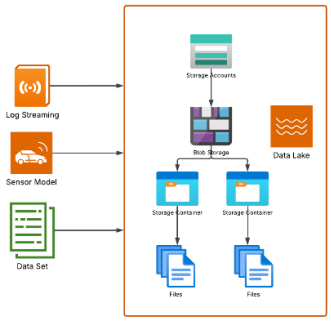
Here you have data coming in at high velocities and also in large numbers.

There are various factors that govern the pricing when it comes to Azure storage. One aspect is the storage cost.

Data storage prices pay-as-you-go	Premium	Hot	Cool	Cold	Archive
First 50 terabyte (TB)/month	$0.15 per GB	$0.0208 per GB	$0.0152 per GB	$0.0036 per GB	$0.00099 per GB
Next 450 TB/month	$0.15 per GB	$0.0200 per GB	$0.0152 per GB	$0.0036 per GB	$0.00099 per GB
Over 500 TB/month	$0.15 per GB	$0.0192 per GB	$0.0152 per GB	$0.0036 per GB	$0.00099 per GB

Reference - https://azure.microsoft.com/en-us/pricing/details/storage/blobs/

A company would want to monitor their storage costs.



Let's say these batch of files have been processed.

We still want to retain the files. But at the moment, they are not required for viewing or processing.

We want to see if there is any way to save on storage cost.

We have to think about saving costs because we might be receiving a large batch of files every hour.

We can change the access tier for the objects.

Hot

This is the default tier for objects. Here this is optimized for objects that are accessed frequently.

Cool

This is ideal for objects that are infrequently accessed. An object can be set to the Cool Access tier. Here the object needs to be stored for a minimum of 30 days.

Here the storage costs are lower when compared with the Hot access tier, but the access costs are higher.

Cold

This is ideal for objects that are rarely accessed or modified, but you still need access to them. An object can be set to the Cool Access tier. Here the object needs to be stored for a minimum of 90 days.

Here the storage costs are lower when compared with the Cool access tier, but the access costs are higher.

Archive

This is ideal for objects that are rarely accessed. And if you need to access them, you don't mind waiting for the data to be restored first.

Here the data needs to be stored for a minimum of 180 days.

Operations and data transfer

Operations and data transfer	Premium	Hot	Cool	Cold	Archive
Write operations (per 10,000)[1]	$0.0175	$0.05	$0.10	$0.18	$0.10
List and Create Container Operations (per 10,000)[1]	$0.05	$0.05	$0.05	$0.05	$0.05
Read operations (per 10,000)[1]	$0.0014	$0.004	$0.01	$0.10	$5
Archive High Priority Read (per 10,000)[1]					$50

Azure Blob Storage lifecycle management

Azure Blob Lifecycle management

Sometimes companies would have millions of blobs located within a storage account. To save on storage costs, it would become a cumbersome task to figure out which objects are not being accessed and then transition them to low cost storage tiers.

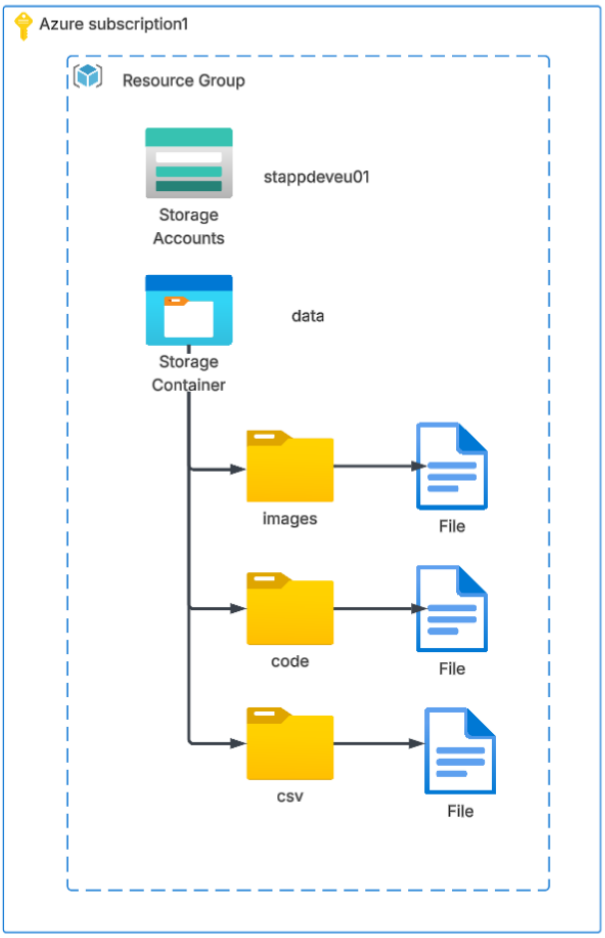
With Lifecycle management, you can define JSON policies that can be used to

a) Transition blobs between hot/cool/cold and archive tiers.

b) Delete current versions, previous versions of snapshots after a time period.

These rules apply to block and append blobs. They don't apply to system containers such as $logs and $web.

These policies run periodically and can take around 24 hours to take effect.



The lifecycle policy is a collection of rules in a JSON document. Each rule has a name. It also has a boolean value for enabled. This can be used to temporarily disable a rule. Then you have the actual definition of the rule.

Filters - This can be used to limit the actions to a subset of blobs within a storage account.

```
{
  "enabled": true,
  "name": "cool-30d-data-code",
  "type": "Lifecycle",
  "definition": {
    "actions": {
      "baseBlob": {
        "tierToCool": {
          "daysAfterModificationGreaterThan": 30
        }
      }
    },
    "filters": {
      "blobTypes": [
        "blockBlob"
      ],
      "prefixMatch": [
        "data/code"
      ]
    }
  }
}
]
```

Actions

1. TierToCool - Set's a blob to the Cool access tier.

2. TierToCold - Set's a blob to the Cold access tier.

3. TierToArchive -Set's a blob to the Archive access tier.

4. enableAutoTierToHotFromCool - If a blob is in the Cool access tier and if its accessed, then it is moved onto the Hot Access tier. This is used in conjunction with the condition of daysAfterLastAccessTimeGreaterThan

5. Delete - This deletes a blob.

If you define more than one action that affects a Blob, then the least expensive one is applied. Delete < tierToArchive < tierToCool

Condition name	Type	Description
daysAfterModificationGreaterThan	Integer	The age in days after the last modified time blob. Applies to actions on a current version of a blob.
daysAfterCreationGreaterThan	Integer	The age in days after the creation time. Applies to actions on the current version of a blob, the previous version of a blob or a blob snapshot.
daysAfterLastAccessTimeGreaterThan	Integer	The age in days after the last access time or in some cases, when the date when the policy was enabled. To learn more, see the Access time tracking section below. Applies to actions on the current version of a blob when access tracking is enabled.
daysAfterLastTierChangeGreaterThan	Integer	The age in days after last blob tier change time. The minimum duration in days that a rehydrated blob is kept in hot, cool or cold tiers before being returned to the archive tier. Applies only to **tierToArchive** actions.

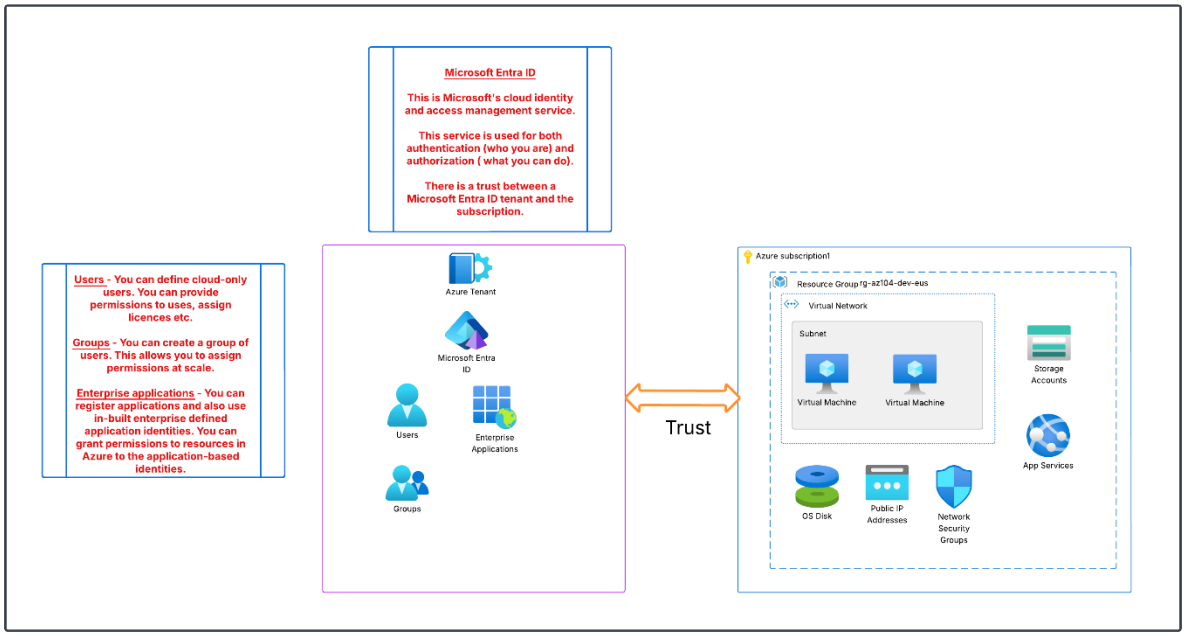
In order to make use of the daysAfterLastAccessTimeGreaterThan run condition, we need to enable Access Time tracking. This does incur additional costs. This adds a blob property of LastAccessTime.

Now if you rehydrate a Blob, it does not update the last modified or last access time property of the blob. Hence defined rules might move blobs back to the archive tier. For this you can add daysAfterLastTierChangeGreaterThan Condition.

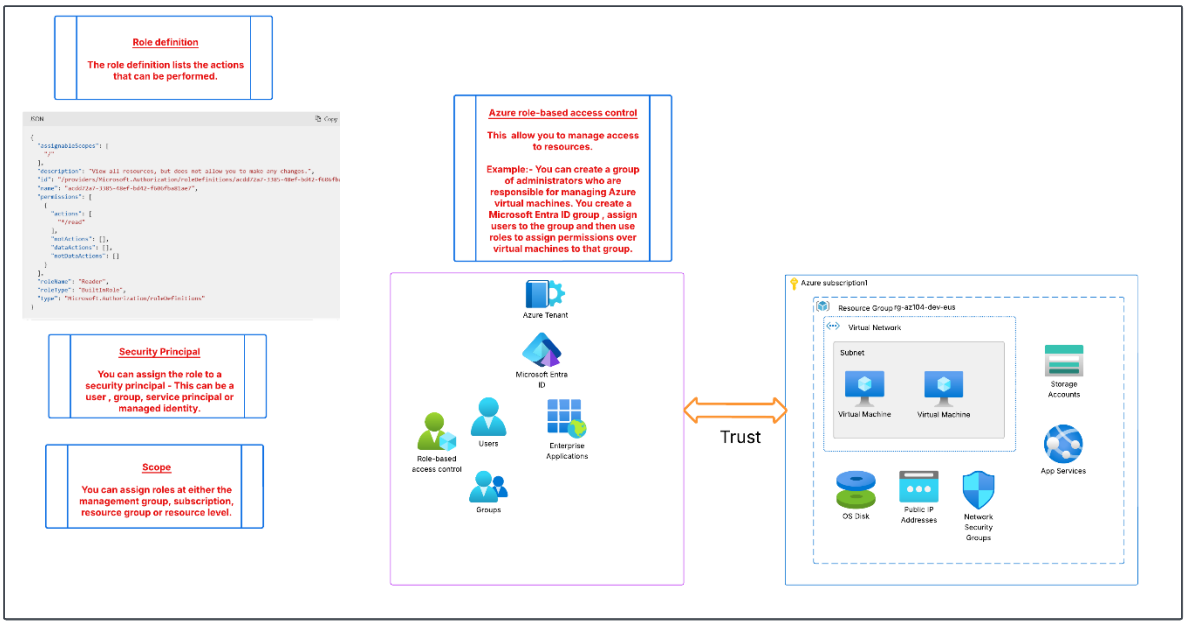
Reference -
https://learn.microsoft.com/en-us/azure/storage/blobs/lifecycle-management-policy-structure

Manage Azure identities and governance

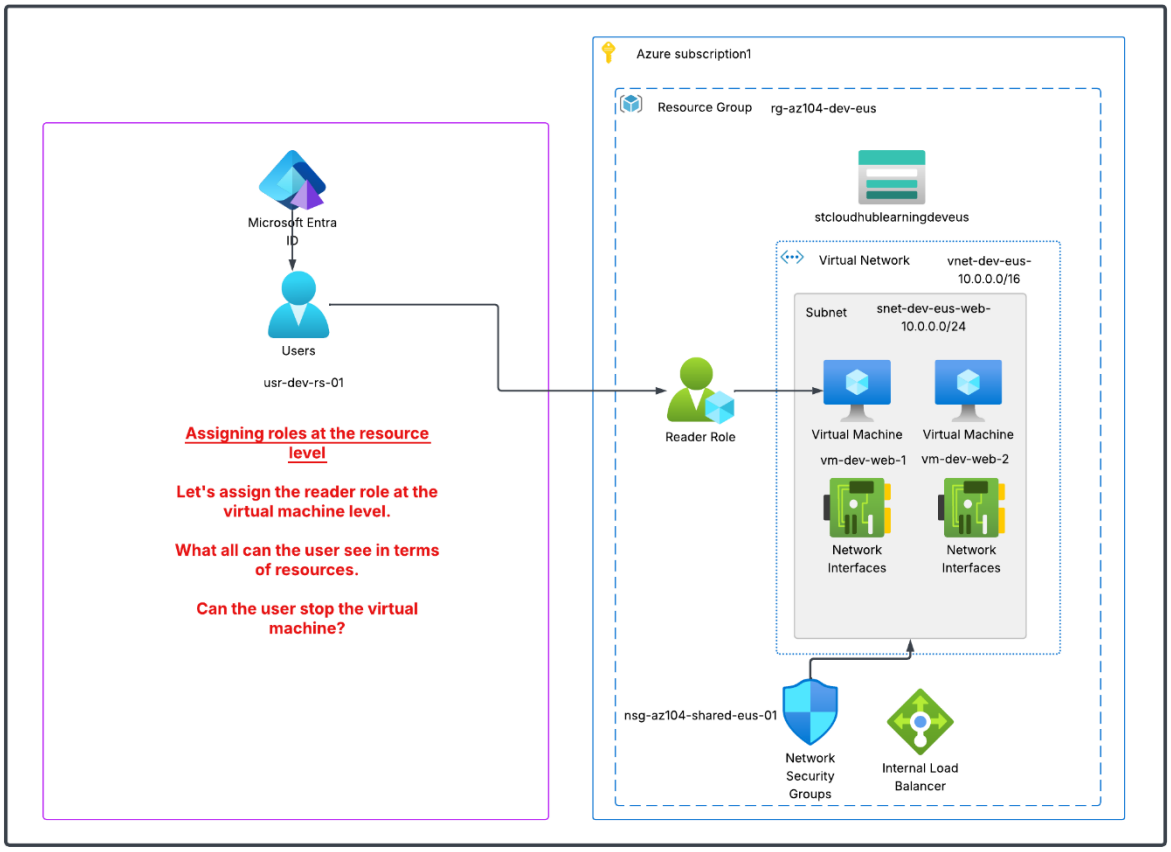
What is Microsoft Entra ID



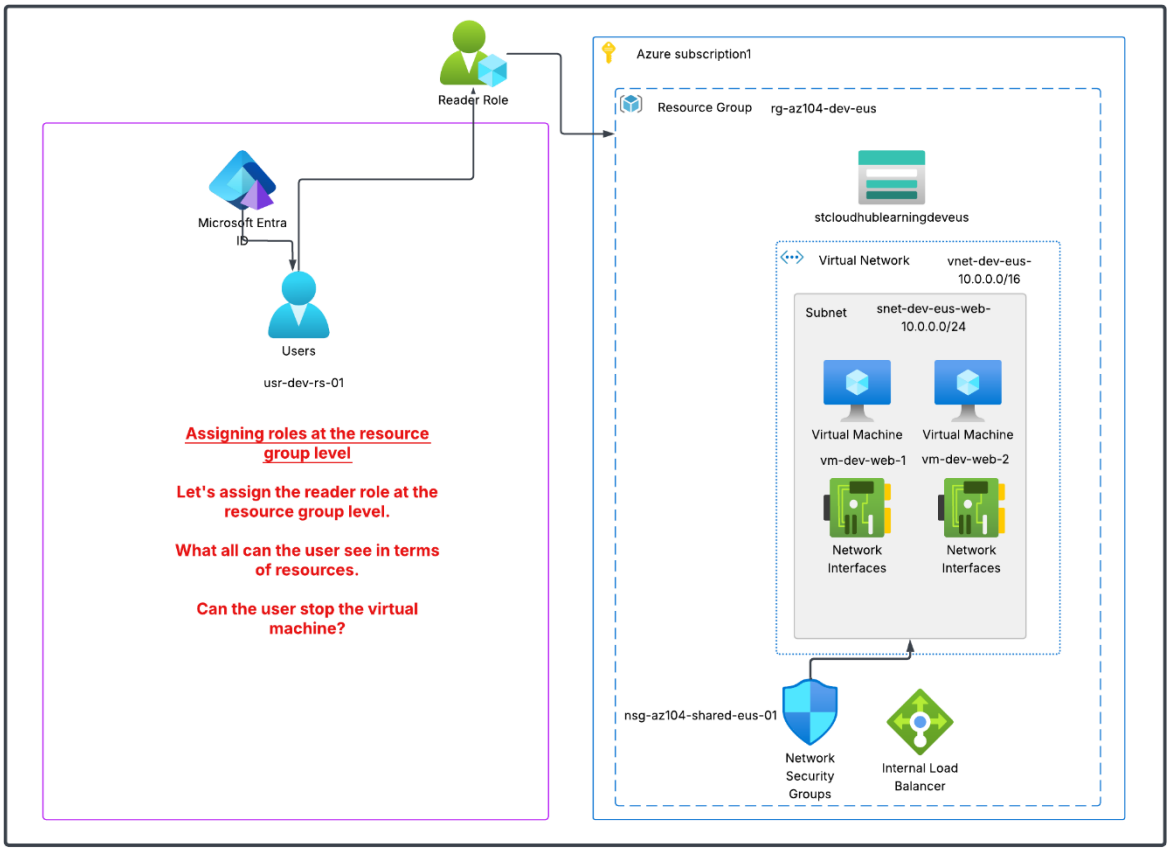
Introduction to Role Based Access Control



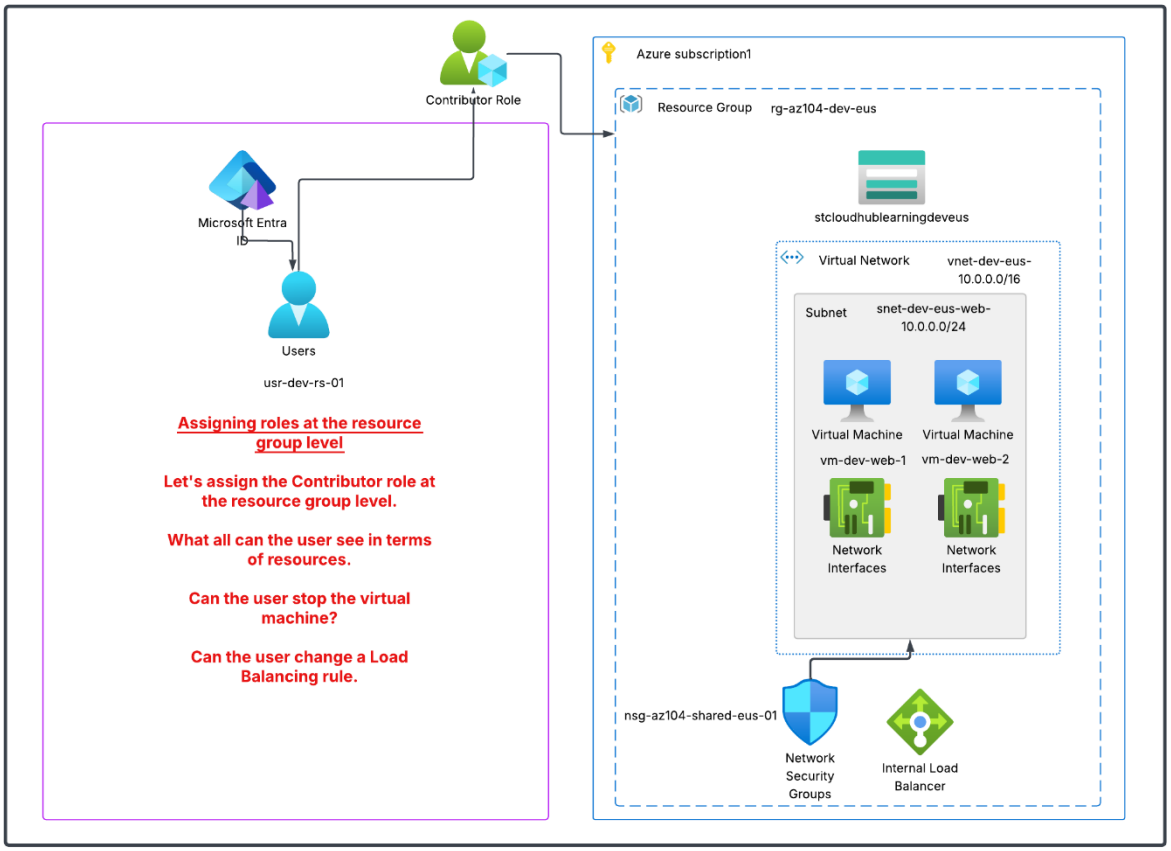
Lab - Role-based assignments - Resource level



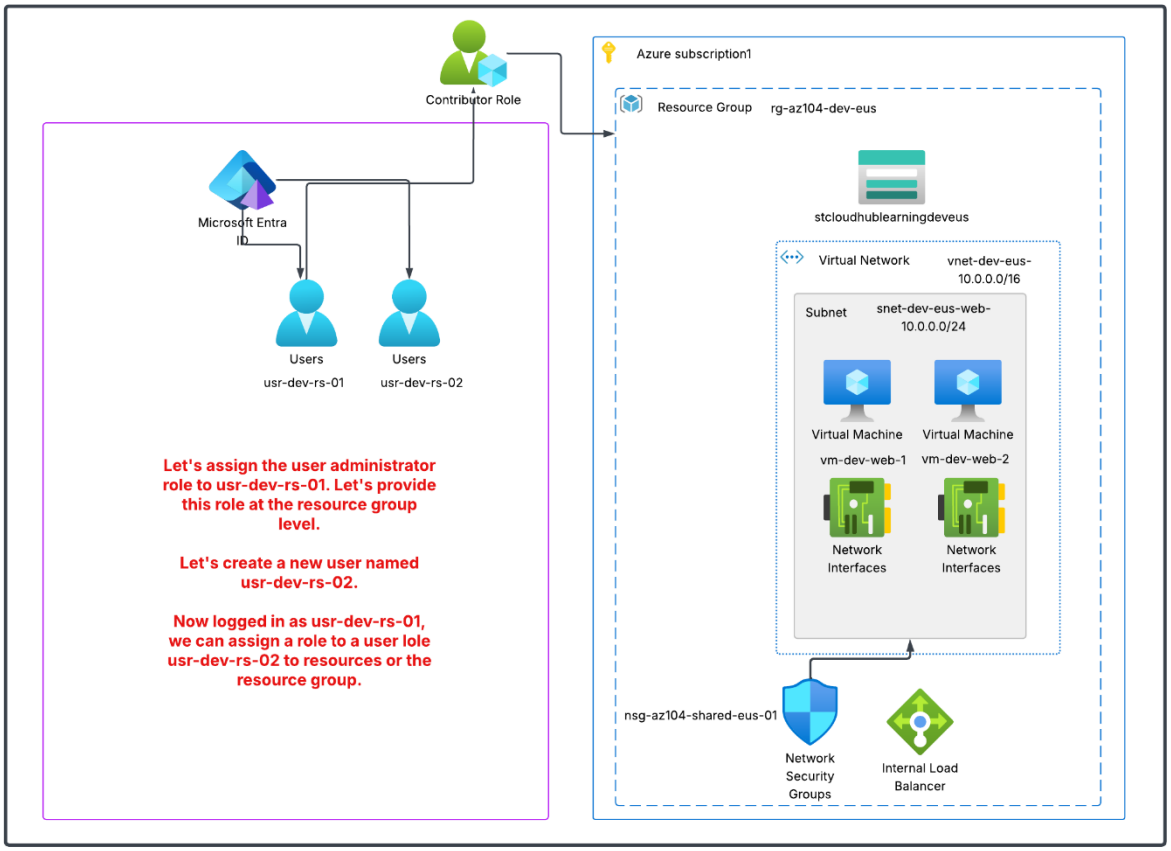
Lab - Role-based assignments - Resource group level



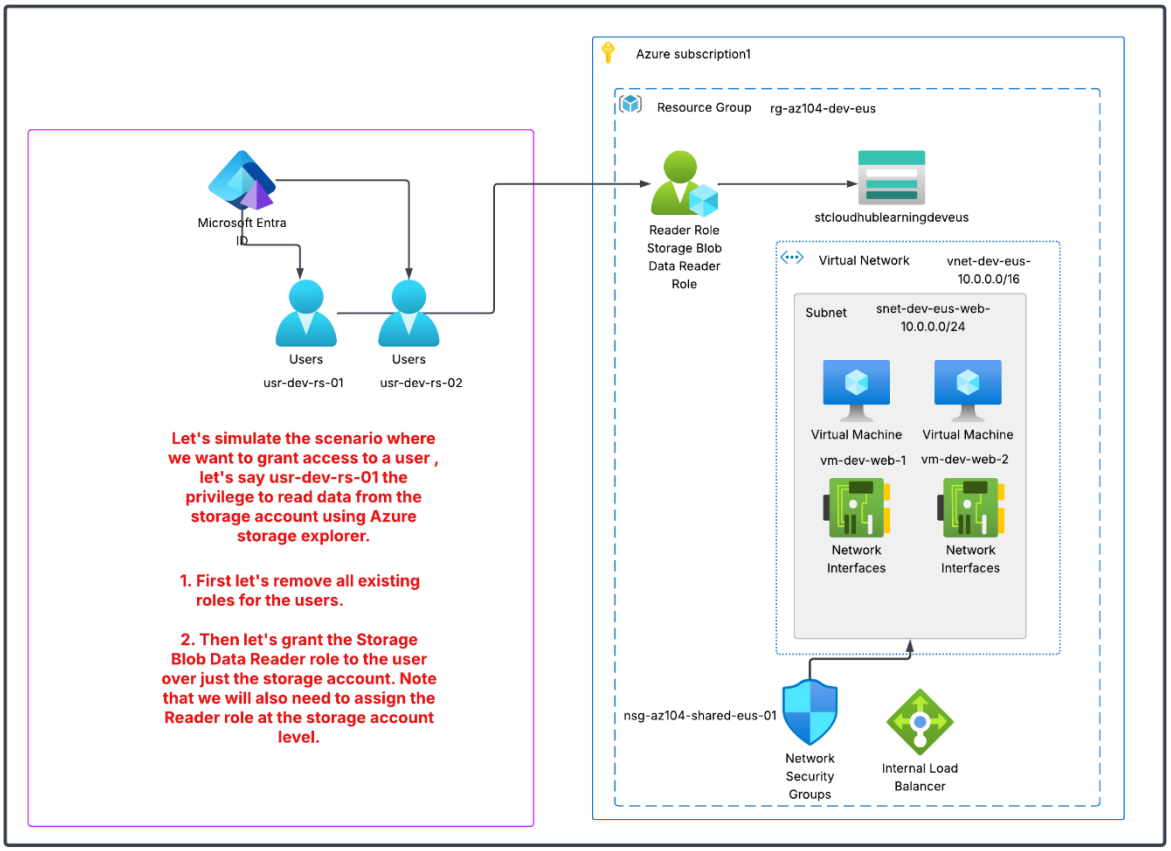
Lab - Role-based assignments - Contributor Role



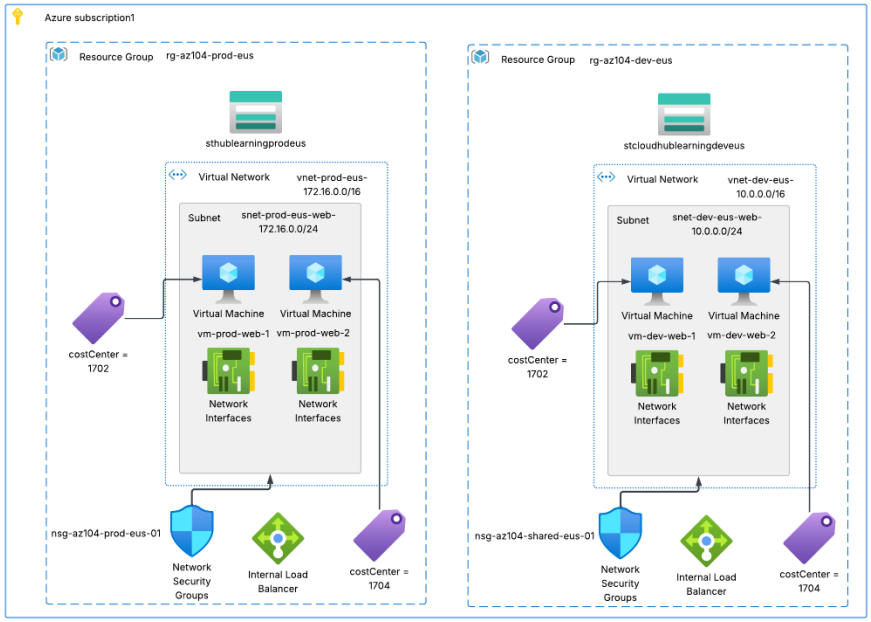
Lab - Role-based assignments - User Access Administrator Role



Lab - Role assignments for Azure Storage Accounts



Resource tags



Resource tags

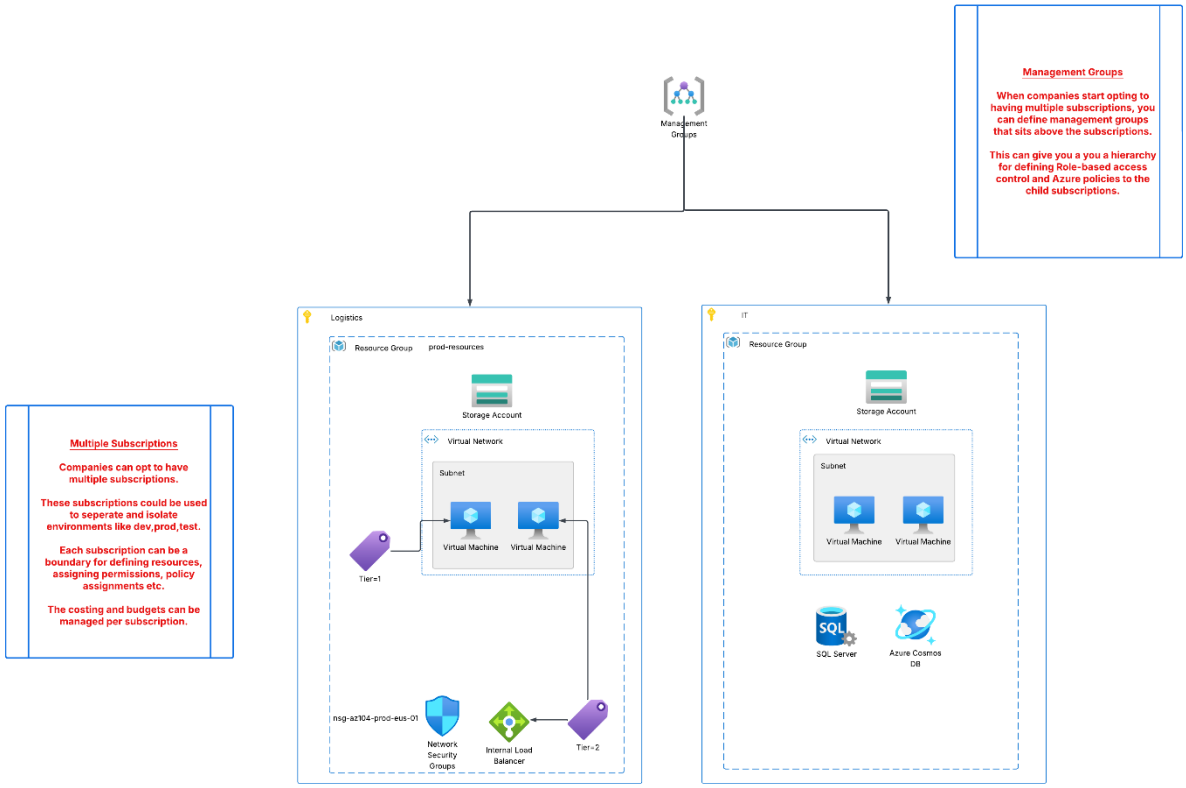
These are key-value pairs that you can assign to resources, resource groups and subscriptions.

These tags can add more business context to your resources. For example, you can have tags suhc as env=dev, env=prod, department=Logistics.

You can filter resources based on tags. You can group and filter spends based on tags.

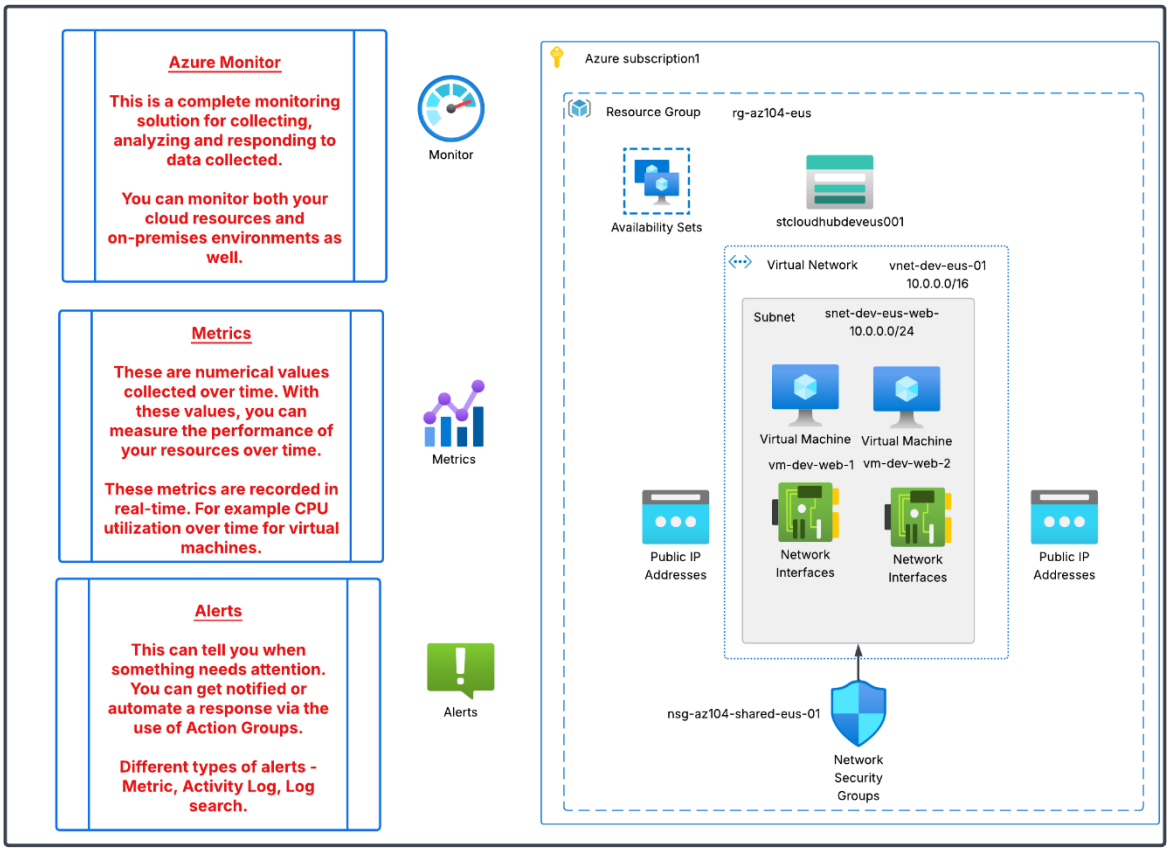
Tags assigned at the resource group level are not inherited by resources in the resource group.

Management Groups

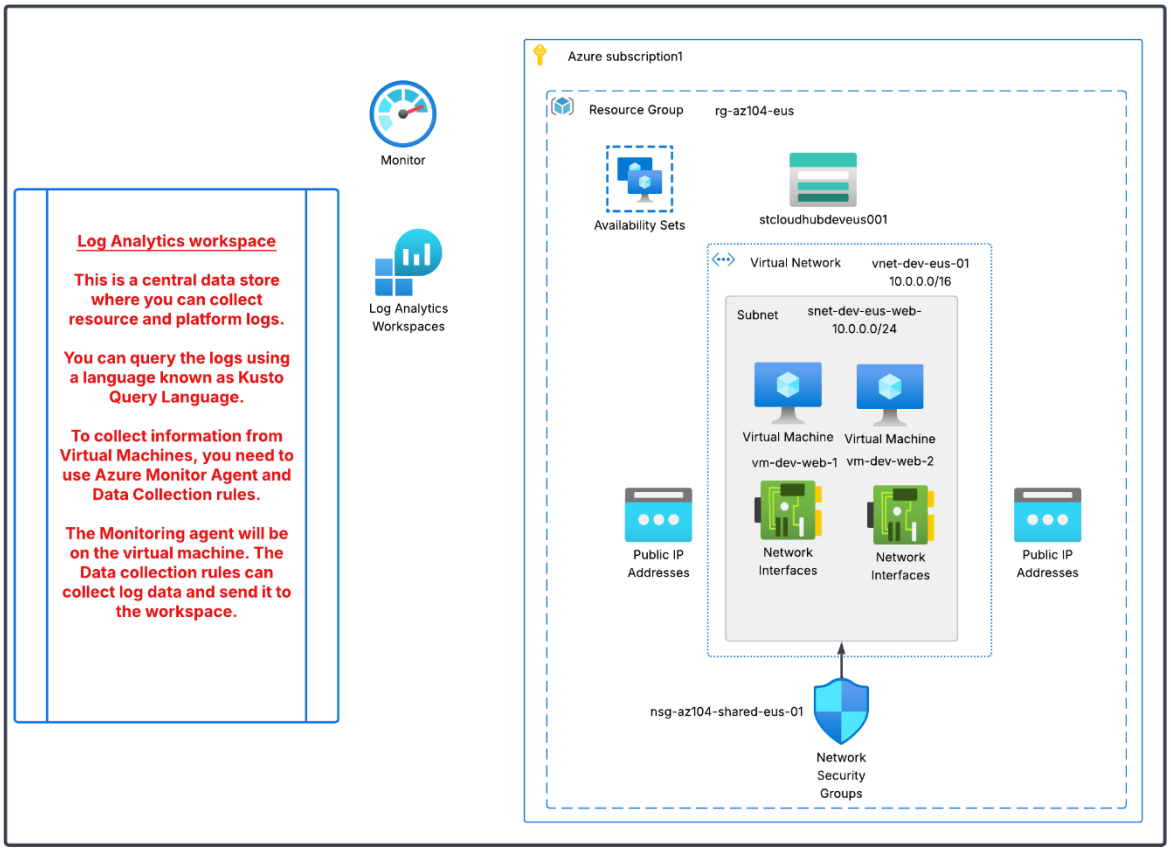


Monitor and maintain Azure resources

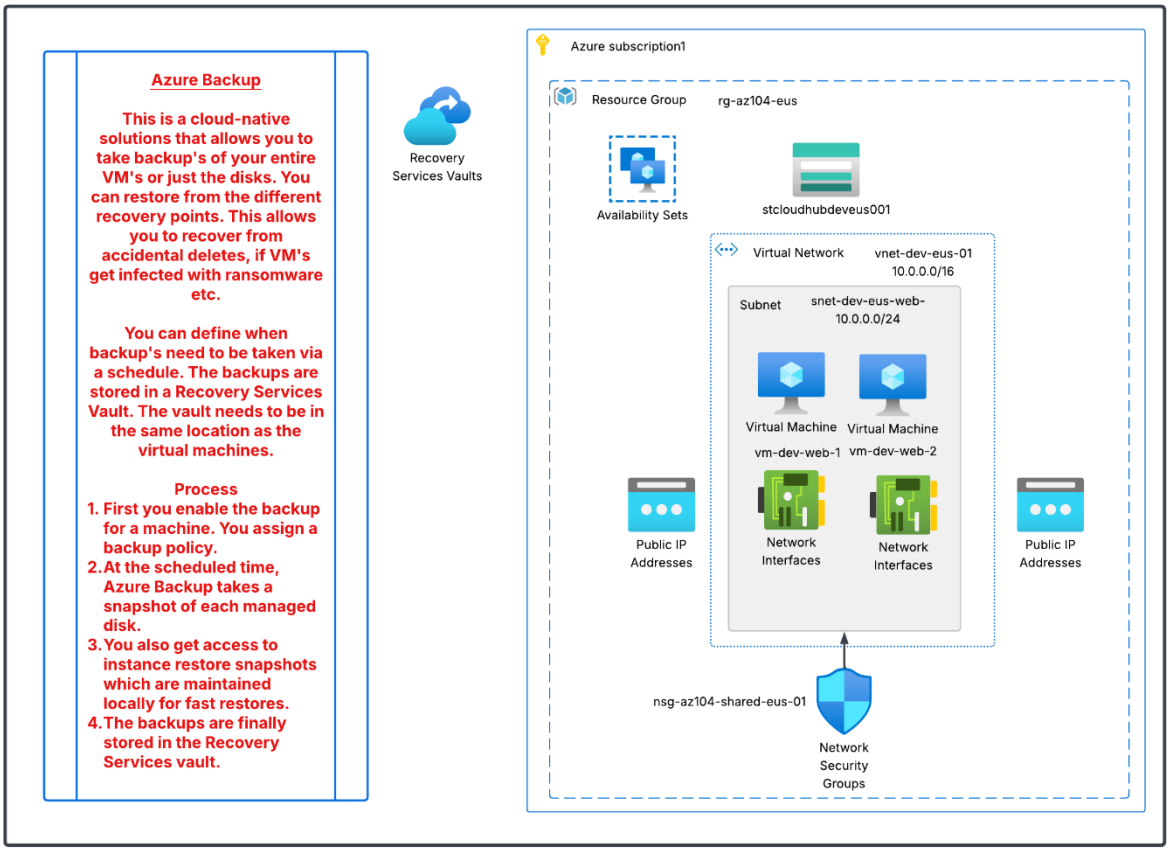
What is the Azure Monitor Service



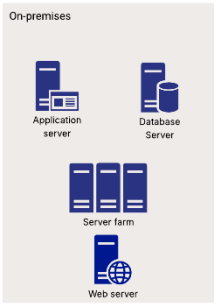
What is a Log Analytics Workspace



What is the Azure Backup feature



Azure Site Recovery



Azure Site Recovery

This is Azure disaster recovery service. This service can replicate workloads from a primary to a secondary location. You can replicate workloads on Azure VM's or from on-premises servers onto Azure.

Again we make use of a Recovery Service Vault for replication.

When compared with the backup process, we get a lower Recovery Point Objective/ Recovery Time Objective.