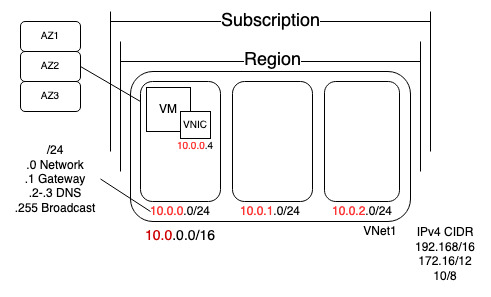
# Virtual Network Basics



So, if we think about the basics of a virtual network, well, it exists within a certain subscription, so that's a boundary. Remember I can create many things within a subscription so we have this boundary. It is the subscription. Now within the subscription I can create things in many many different regions. But here are pick a specific region, maybe it's Switzerland, it could be Germany too, it doesn't matter. Within a particular region, I can create a virtual network.

I've created my Vnet1. Now the whole point is that virtual network will defining by at least one IPv4 CIDR range. So a CIDR is a way to write down a particular network and how many bits make up that network. For example, I can use 192.168/16 and remember this slash 16 is a number of bits that make up the network mask. So it's 32 bits in IPv4 range. There's also 172.16/12 so this could actually go up to something beyond the 16. And then as 10/8. Those are RFC 1918 ranges. They'll still be private IP addresses used only within the virtual network. For Virtual Network I can add 10.0/16 to keep half of the /8 available for future growth. Now within that virtual network, I create subnets. I'm using a portion of that 10 space I'm using hey the 10.0. That I could then break up into subnets. I have to use a portion of that space, so maybe I'm going to make subnet one. Then I use a portion of that, so maybe I'll do 10.0.0/24. I might create a second subnet so I can have another subnet. Subnet 2. Maybe this is 10.0.1/24 so you can see it's just moving along and I'm using whole 8 bits per subnet just to make it simple for me. Now when I create these subnets, remember what we said last week. We lose a set of IP addresses, so whatever this range is. So say that was a slash 24, I always lose five IP addresses. So, in this /24 .0 is the network address. The .1 is always used for the gateway, so Azure is providing the gateway service. Even if resources are in different subnets, they can automatically just talk to each other because Azure is providing that gateway for us. The .2 and the .3 are reserved for DNS purposes. And then .255 in this case, used for broadcast. So, I always lose 5 IP from any virtual network, no matter what that size is. If I create a /29, there's only be 3 usable IP addresses, so 29 would give me 8 addresses, but I lose 5 so it could be 3 usable. /29 becomes the smallest possible in Azure, because anything less than that there won't be enough even to give it the reserved IP addresses it wants.

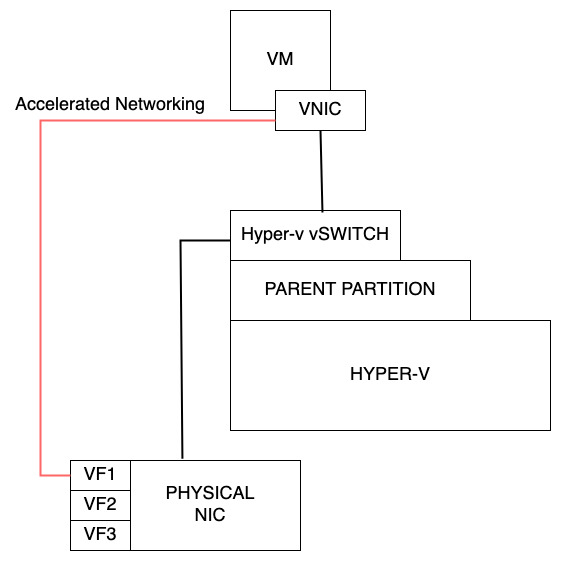
We have those availability zones in most regions AZ1, AZ2. AZ3 and subnet span them. There's no alignment of a subnet to a particular availability zone. They don’t care, they can just crossover those and then I would create resources.

Within these subnets I can create a VM and what I say I create a VM in the subnet, it's not really the VM. What really happens is that VM has a Virtual Nic and it’s the NIC that connects and is bound to a particular subnet in a particular virtual network.

So this is all virtual network. It's bound to a particular region, cannot span regions, cannot span subscriptions. That's just the way these resources are constructed.

# VM NIC

So, there is the NIC. The IP address from a virtual machine always comes from the Azure fabric. This will be thought of within the guest operating system as DHCP. Now I can reserve a particular IP address from the subnet. If I have a domain controller or a SQL Server I can absolutely configure it so it always gets the same IP, but the guest operating system it thinks it's using DHCP. I can have multiple NICs, connected to a VM. You can have multiple NICs but they have to be connected to the same virtual network. So they could be connected to different subnets, absolutely, but it could not have a Nic connected to a subnet in a different virtual network. Every subnet it connects to has to be in the same virtual network. And if you think about the use of multiple network cards, that's useful in a physical world. In a physical world, there's pieces of wire attached, that connect to physical hubs that connect to physical other pieces of wire, copper, fiber, whatever it is. So it actually separates up my connectivity. But in Azure is all software defined networking. The use of multiple NICs becomes a lot less useful in a software defined networking world. It's better to use network security groups and rules to control traffic. I can use routing to control things.



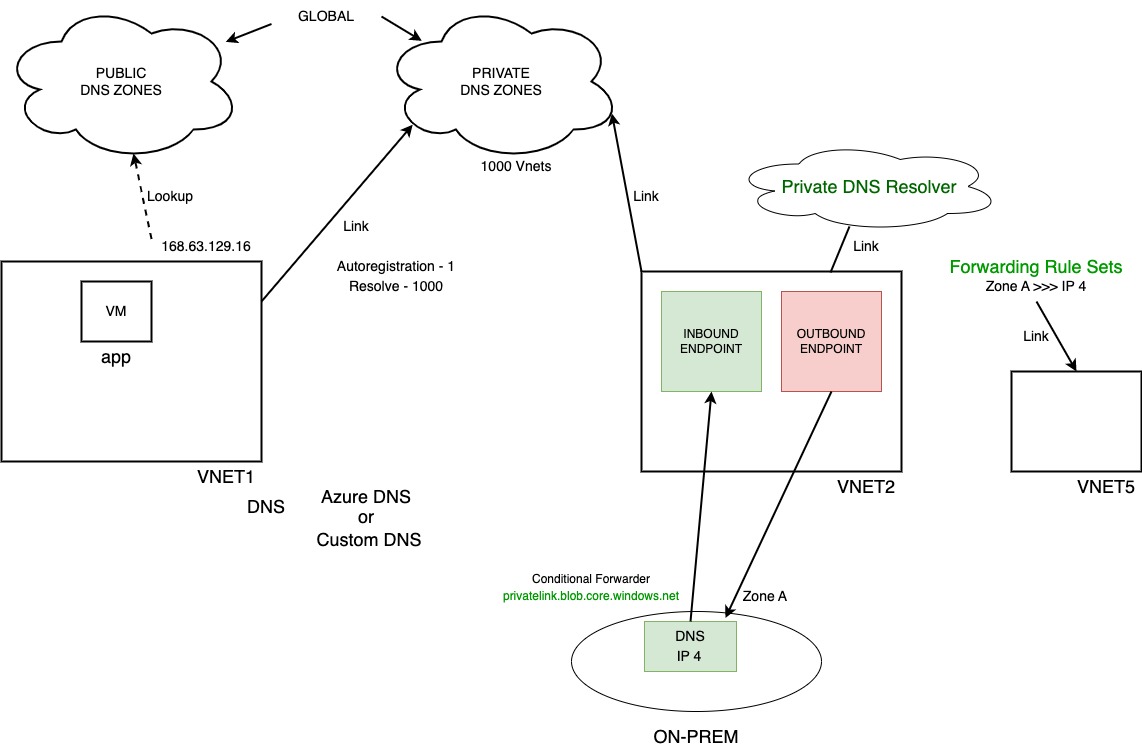
Going back to NIC and VMs, many VM types also support accelerated networking. So what is accelerated networking? When I talk about a virtual machine, behind it there's some bare metal. So there's some bare metal hardware I could think about, and then we're running a hypervisor. In this case Hyper-V. The hypervisors really concerned with CPU and memory. And then there's this parent partition. And the parent partition because of the flexibility of acquiring different types of driver and I don't want those in the hypervisor directly, this parent partition runs. And connects to other types of hardware. So if I think about the network interface card, the Nic, the physical Nic on this bare metal piece of hardware, when it's actually the parent partition that is bound to that Nic and it has an hyper-V switch that enables all of the connectivity. So the hyper-V switch actually connects to the Nic and then when I create my virtual machine and it has that virtual Nic, it connects to the virtual switch.

The virtual switch is where we have all these virtual filtering platform components that do the various bits of routing and so on. Great stuff, but it adds a certain amount of latency.

Well, what we can also do is there's technologies like SR-IOV that exist in Hyper V and in the hardware. Network cards expose things called virtual functions. They really can expose themselves like their actual physical adapters. So what accelerated networking does is, instead of it going via the hyper V switch, it gets bound to a particular virtual function. It bypasses the Hyper-V switch. It basically just reduces latency, improves performance. Not every VM SKU supports it, but many of them do. So if I see the option there for accelerated networking, go ahead and turn it on.

I can have multiple IP configurations per NIC, both private and public.

# DNS in Azure

So we get the DNS in Azure. A virtual network is configured to use certain DNS servers. Now there is a built in Azure DNS. That just fingers by default will use. So it's this special IP address within the network that will just get leveraged by it. It's 168.63.129.16 and that will provide IP resolution or I can define my own. So for the DNS I have a choice, there's Azure DNS or I can configure custom my own DNS servers. If I use the Azure DNS, what it's really pointing to is this special IP 168.63.129.16 or I just have my own ones. And then what that does is obviously looking at records. Now there's certain just built in records.

That is really two types of DNS I can think about. Well, there's public DNS. This is where I go online and I look up avaloq.com. Well those zones have to be available to the Internet and I can create Azure DNS zones so this will can go and look up against public zones so I can create those in Azure. Obviously it has to be authoritative for that zone. But I can create IPv4 and IPv6 host records. I can also create alias so Cname records.

They are global resources. So when I think about these, these can be used in any region anywhere. So that would be public facing. Then we also have the concept of private. I have some zone that's going to have records in it that I don't want to resolve from the Internet. I just want them to be resolvable from one or more virtual networks. And so, what I can do is I can create these private dns zones and then I can link it to a virtual network.

Now when I link it, it's actually two different types of things I can do. When I create resources, it gets an IP address dynamically and maybe I want its host name to get registered. So there's something when I do this link I can configure the link to use auto registration.

Now from a VNet it can have one private DNS zone they will auto register with. So when these names. It will go and create a record. So if this was avaloq.net and this was called app1. It would create a record called app1.avaloq.net with its IP address it got.

It can also be linked for resolving up to 1000 private DNS zones. So avaloq.net, avaloq.org, all these internal things. Let’s talk about storage resources like BLOB. They have a URL that could be a private DNS zone as well. That's a great use for actually using a private DNS, and each private DNS zone can be linked to up to 1000 VNet. I can have one zone, but I could also link it to my vnet 2. It could get linked to the same so they have the same resolution for example, for whatever the content is within there.

One of the challenges you can get with DNS is dangling. Normally when I create a service in Azure it has a name, but the name is not very pleasant. It might be something.core.windows.net, whatever that might be. So that's the Azure name. I don't want to give that to my customers.

So, I would probably on my DNS I might create app.avaloq.com and I would point that so I could create like a cname record in my DNS to this Azure service name. So far so good.

Time passes and I retire this service. I delete it. But I forget to delete the alias, so right now there's alias resolves the same that doesn't exist. Not a problem until Mr. Bad Guy comes along.

And because this is just an Azure name, Bad guy creates a service called the same name as I did. Now my cname points to their service because they just took over the name because I forgot this was a dangling DNS. That's a risk. There were somethings being built in to Azure to help resolve this, but that's a real problem. There were bad actors out there that are going looking for dangling DNS and they'll try and create the name that this used to point to and then try and get people to use this because you're gonna trust it and it's not trustworthy anymore.

So my whole point here is one of the nice things that I can do with the Azure public DNS. It's rather than being an alias to a name. They create a link to the application when I create an alias record in Azure DNS services. So it automatically points to the resource. If I deleted the Azure service, the alias would just stop working. It's not dangling, it can't be taken over by someone else.

Now if I jump over to my private DNS zones. So remember these would not resolve over the Internet. The whole point of private DNS zone is I would link them. But notice what I have here is I have this private dns zone privatelink.blob.core.windows.net and the records it added it for me so I didn't have to manually create this, it just resolved it. This is great. Azure provides this DNS server for me.

But I still have maybe my on-premises network, so give yourself a bit more space. So I've got my on Prem network. It would like to be able to use this private DNS zone, but it can't because it's only available via this 168.63.129.16. So the only thing I could do is to add a VM with a DNS forwarder in here and point to that, which is an option. I can definitely do that, but my other option now is there's this service.

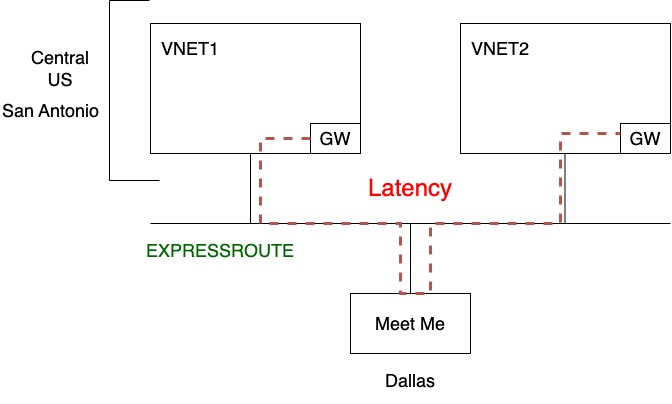
The private DNS resolver and it does two different things. When I create it, it links to a specific vnet. Now in that specific vnet, there's two things it can do. Inbound and it can do outbound. I don't have to use both, but if I do, I need a subnet for each. The inbound endpoint is just an IP address and why this is nice is that now other networks could point to that IP address. So from on Prem, my DNS servers could conditionally forward for blob.privatelink.core.windows.net. So I don't have to manually create records, I can just point to the private DNS resolvers inbound endpoint. So from my DNS on-prem I could have a conditional forwarder that now says hey for BLOB.privatelink.core.windows.net go to IP1. I've got site to site VPN or private link private peering. There's an IP path, it would now use it.

There's also a second challenge that sometimes from Azure DNS I want to conditionally forward to a zone that's hosted on a custom DNS server, and I can't do that. Well, now I can. In private DNS I can create forwarding rule sets. Which I can say hey to get to zone A you have to talk to DNS servers IP 4. Now it has to use an outbound endpoint. Why? Private DNS was over. Is this floating service out there in Azure? It has no actual network. It needs to go and look up these IP addresses for you. It's not just going to tell you go and talk to IP4, it will go and talk to IP 4, resolve the record and return you the response. So it needs a network path to get to IP4. That's what the outbound endpoint does. It uses this to go and IP talk to this DNS server to do the lookup to return you the response. Now once I've set this up I could use this the same VNet 5. I can link this. This rule set can be linked to other virtual networks this bring. So that's what a private DNS was overdue. So it's two things. One. It's a target to my DNS. Can now query against records in private DNS zones. But also, I've got other DNS zones on custom DNS so servers. I can now tell Azure DNS go and talk to this DNS server to be able to look up records for me. So that's the private DNS resolver.

# Connecting Virtual Networks

Remember, if I have multiple subscriptions, if I'm using multiple regions, those two things are the boundary of a virtual network. It cannot span those things. So I may end up with multiple virtual networks. In the past there were ways to connect virtual networks, but those ways, they were ugly.

**ExpressRoute**

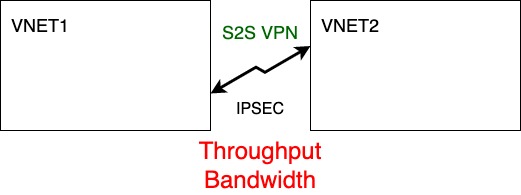
****

So this is the first example. I have got two virtual networks. So I've got VNet1. And I got VNet2. Let's say they were actually in the same region, they don't have to be. This doesn't have to be the case for this, but it helps show why this is bad. Let's say this was in South Central US, which remember means they're in San Antonio.

And let's say in our example we want to use our ExpressRoute as a private connection. We're going to talk about this, but the way ExpressRoute actually works is there's a Microsoft backbone network. Which these basically get connected to because they're going to have a gateway. In each of them. And then what happens is they're saying called meet me peering point. And normally this is where I would then connect my network to it, which would then connect to here. I want to connect my 2 VNets together by connecting them to the same ExpressRoute circuit. They're in the same data center. The path the traffic will flow is to the meet me. This meet me might be in Dallas. So even though the servers that being spoken to each other might be sitting across the data center, they can see each other, they can wave. If I connect them to the same circuit to connect the VNets together. It's going to route via the Meet Me location, which could be hundreds of miles away. The latency will be horrible. I don't want to do that.

This is a bad scenario because latency. I'm wasting the throughput of the circuit. This is just a horrible thing.

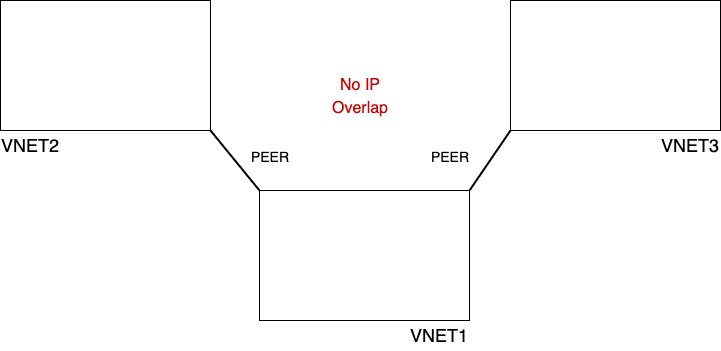
**VPN Site 2 Site**

****

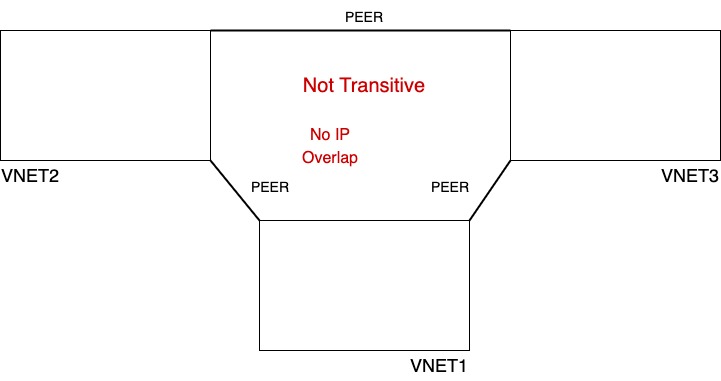
OK, we said express route, that's not a good fit. What about a site to site VPN? I have the exactly the same scenario. I have exactly the same. I've gateways but this time they're site to site VPN gateways and I'm going to establish a site to site VPN to connect them together. Well, remember how does VPN work? Encrypt the traffic expecting it to go over the Internet. Now these wouldn't go over the Internet because they're both in Azure. Even though it might be a public IP, it would stay on the backbone network, but it's still using Ipsec. Ipsec is expensive in terms of computational resources to do the encryption and decryption. So what here is throughput. It's going to be bad, so this is not that great either. So the amount of bandwidth is going to be limited. By whatever the SKU is of the gateway, because that's how big it's CPUs are, that's how much traffic it can do. So I don't want to do express route to connect VNets together. I don't want to do site site VPN to connect virtual networks together.

**VNet Peering**

So what do I do? What is my answer? VNet peering. Vnet peering enables me to, on the Microsoft backbone, just connect virtual networks together. They cannot have overlapping IP ranges, but they can be in the same or different regions, it doesn’t care. There is a small today ingress egress charge, so I pay for the data going out, I pay for the data coming in so that there is a fee that I'm going to pay for that. And again, the IP address space cannot overlap.

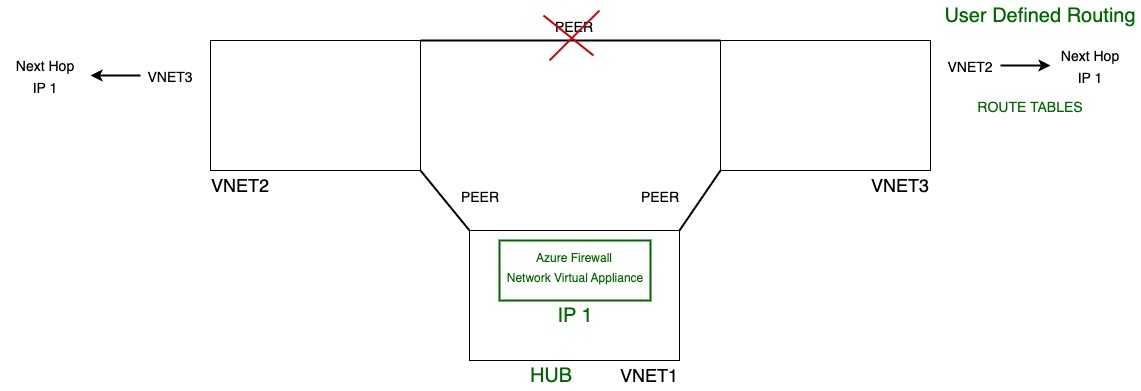


This is VNet 2 and I'm just going to pair them. This is the best option to connect things together. But again, when I'm doing this peering, what I cannot have is no IP overlap. They cannot use the same IP addresses.



But I could obviously connect multiple V Nets together. So we have this vnet over here as well. So we'll call this VNet3. Can these talk? This is a very common model hub and spoke. Here's the hub. These are spokes. Can the spokes talk? No, they can't. It's not transitive. So if I wanted these to be able to talk, I would have to explicitly add. Peering relationship between them. That's how I could make them actually be able to communicate with each other now. It can span subscriptions, it can span Azure AD tenants, but it's not transitive.

But they can be. What does that mean? By default.

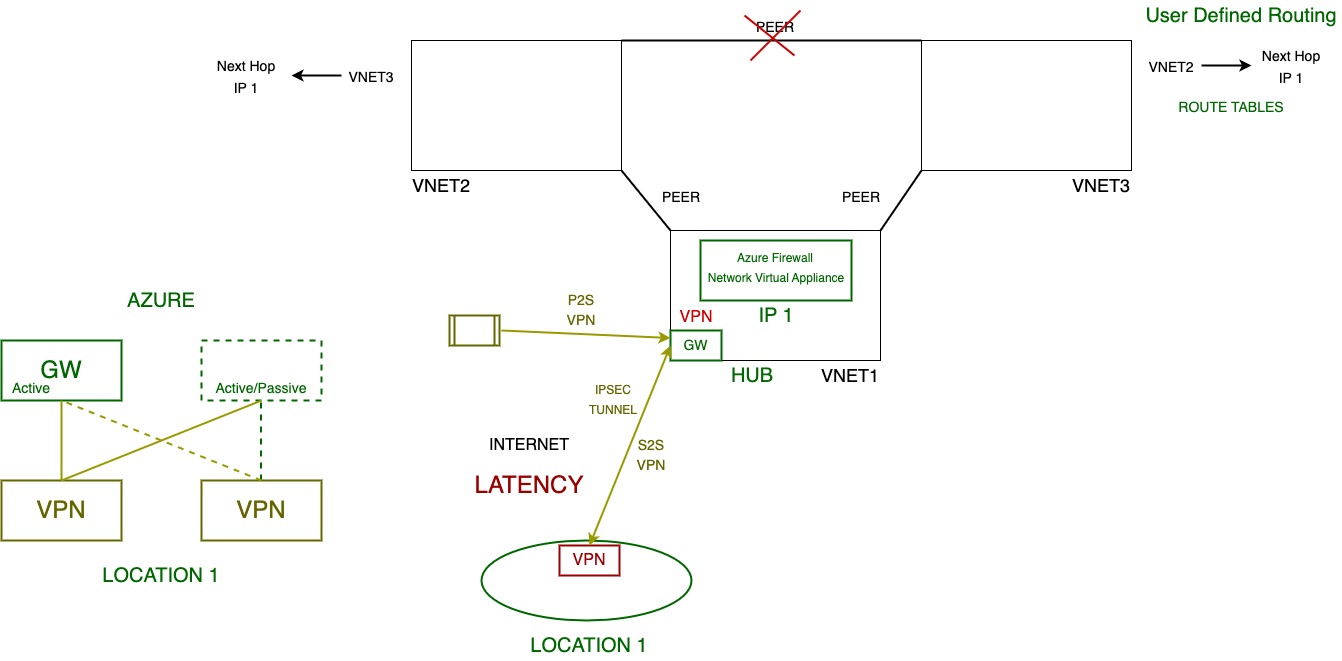


They they have no way of talking via the hub. This is the hub. Well, they're both connected to the hub. There's nothing in the hub that will take a packet from here and send it to the here for them. This VNet2 has no reason to think, oh, I should send it to the hub VNet1 to be able to get to that. What I can do is let's take out this peer I could add in this hub network. I could add maybe Azure firewall or just a network virtual appliance that does forwarding. That can accept the packet and send it somewhere else. If you want to get to VNet 2 know this, whether it's a network virtual appliance or Azure firewall, it's going to have an IP. So let's say this is IP 1. I need to tell VNet 3, hey, if you're trying to get to VNet 2 your next hop is IP 1. So the place to send your packet is IP 1, and likewise this one needs to say hey, if you're trying to get to vnet 3, your next hop is IP 1 as well. The next hop doesn't have to be in the same subnet, even within the same virtual network, there's really nice things that we can do, because it's software defined network. Normally I could never do that. The next hop would have to be in the same subnet. Doesn't have to be the case here. Talking about next hop, Is this appliance over here now. How am I doing this? How am I defining to say, hey, when you're trying to talk to this IP space, this is your next hop. Well, the solution is called User Defined Routing. So what we're doing is we're creating route tables. So this enables me to override, or just add to the default system routes. So this is how we can achieve making them transitive. I have to have something in a network they both connect to that's capable of forwarding packets. And then tell the virtual network and the subnets to actually get to this place, go to this next hop and it will then tell you where to go. So this is a a really powerful capability. Now I don't have to add this whole spider web of peering relationships.

# Connecting to On-Premises

**VPN**

So I think about connecting to on premises. We have our own locations as well. Maybe we've moved everything to Azure, but likely we have data centers. So, many Azure services have external Internet facing endpoints that I could use. Yes, I could go and connect to instance that has public IP of a VM. Yes, I could go and connect to my storage account with its public IP addresses. But often I don't want to do that. I'd rather talk to some private connectivity. There are different ways I can achieve that. So the the first ways of doing that is I could think about this a point to site VPN. So it's going to connect to specific device to a virtual network. There's also site to site VPN where it's connecting a network to another virtual network. And there were different types of gateway sizes that have a different number of support for how many networks can connect to it, if it can learn dynamically the IP ranges, there's different sets of functionality around that.



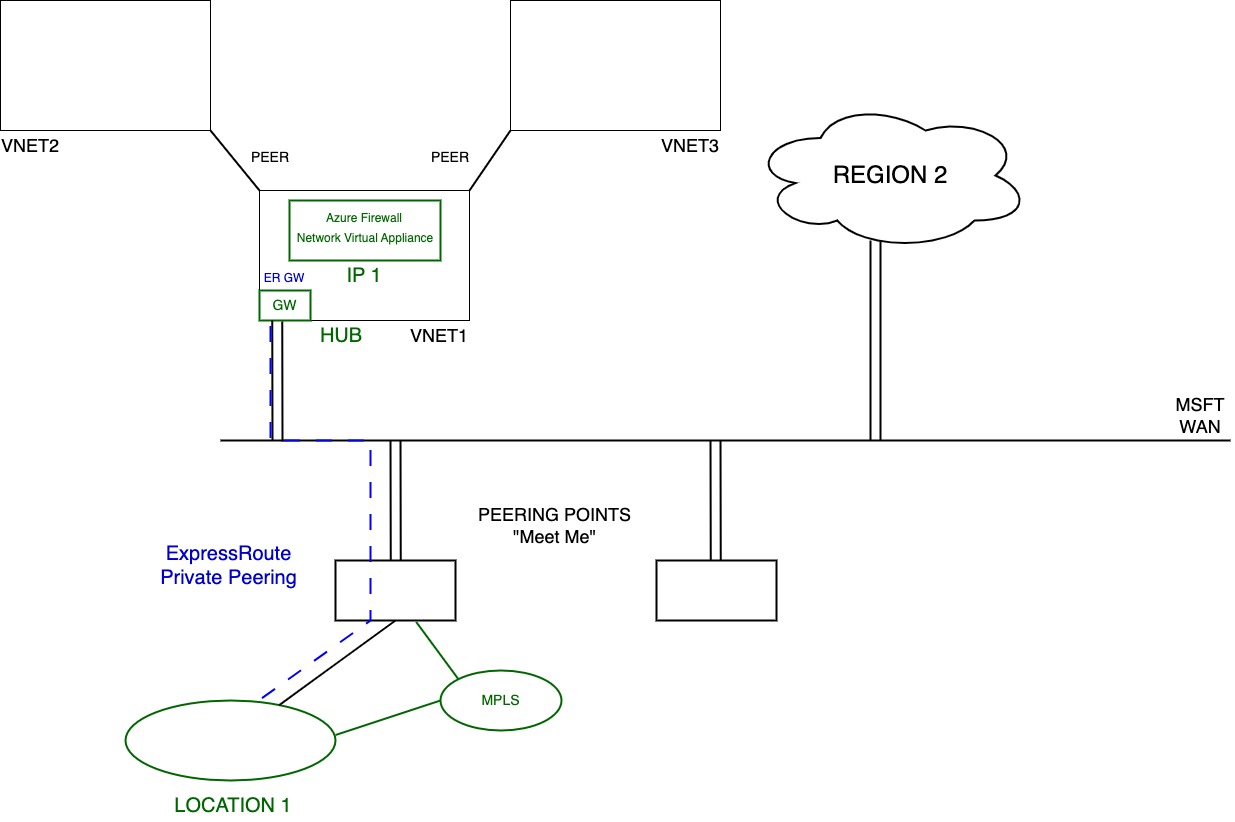
We had this actual picture here. But now I also have my on-prem, I've got my Location 1. Now it's different things I could do. Now certainly within here I'm going to have a gateway. This is my hub, so I'm going to use this to talk to other stuff. Now, the first type of other stuff that I might want to use would be the VPN. So, to use VPN on premises I also have to have some kind of VPN device. And what we're going over, remember, is the Internet. There's no special private wires between them. It's the Internet and what we're basically establishing is a tunnel. So we have this IP SEC tunnel over the Internet. Now this is a site to site. A point to site would just be this machine and only this machine. So if I had individual contractors at home, they could do a point to site and this works. I mean it works fine. There's different SKU's have different amounts of bandwidth they support. I think it's 2.3 gigabits per second is the highest per tunnel. A single gateway can support multiple tunnels to it. Now the aggregate it might be 10 gigabits per second for the high end SKUs. But a particular IP SEC can only be serviced by particular core virtual CPU. So any tunnel is restricted based on the capabilities of the processor.

Now it does actually open up a few different architectures. If I think about high availability now, I drew one location. So let's say, OK, I've got my VPN. And then we've got the Azure side, so we've got the Azure Gateway and it could absolutely be just this. It could be. Azure actually always deploys 2 but this one might be standby. So this is always active. This might be active or it might be standby. So my next option I could do is from on-prem I'll connect to this one as well. So over redundant connection. If there was some problem on this, these are VMs behind the scenes saying it happened to this one, it could fail over to this one. Fantastic. Maybe I have multiple VPN devices on my side? Could it be different locations technically? And maybe my path here is. Well, actually I'll connect to this one. From both my locations. Well, maybe I connect to all of them. So, there's different levels of resiliency I could do. But now this is very resilient. This resilient against the failure on one of the Azure side, it's resilient against the failure on my side.

But it's coming over the Internet and going over the Internet means, well, the latency between these connections it could vary, because it's the Internet. It's not going to be a set standard latency, it's just there's going to be some variation.

**ExpressRoute**

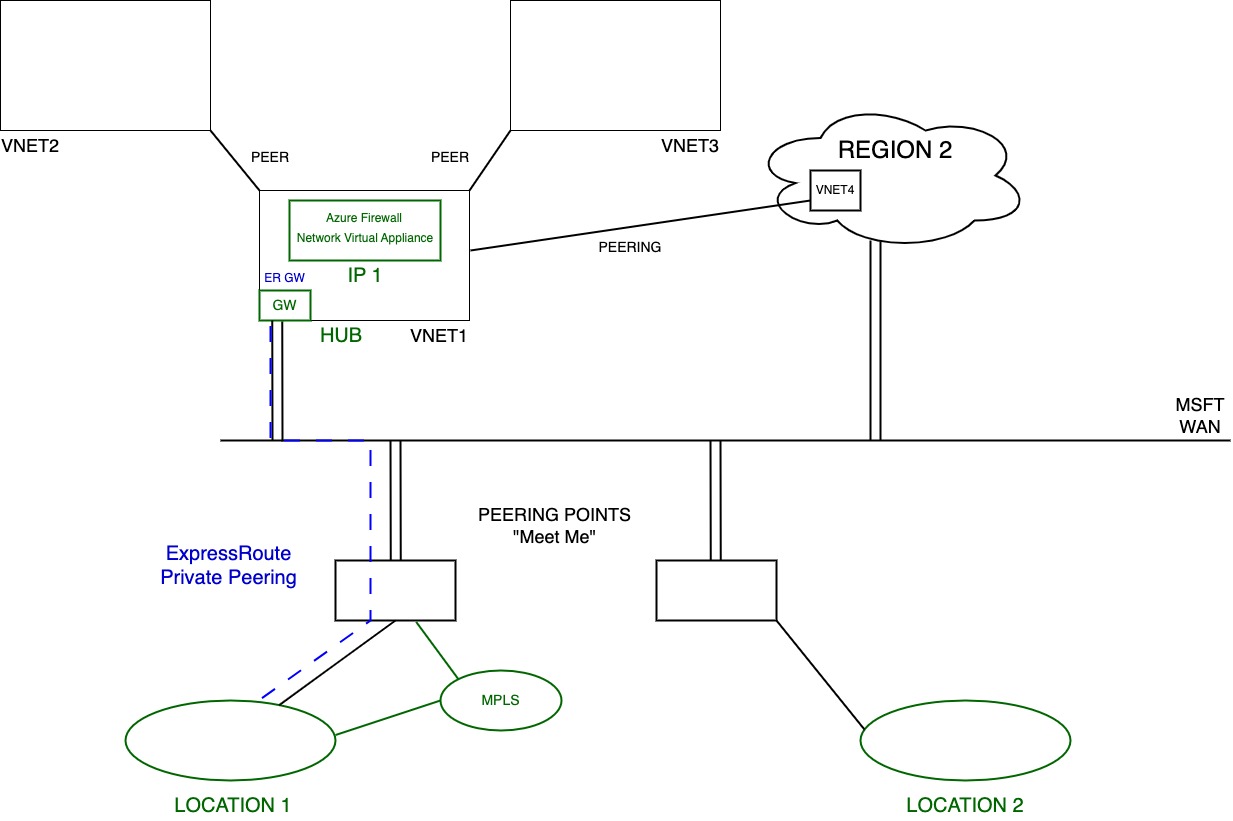
So next thing we can do is express route. There's a big Microsoft backbone network, and what Express route does is there's different types of peering. So there's Microsoft peering. Microsoft peering lets me via particular route tables. I can say, hey, I want to offer Microsoft 365. That's a bad example typically, but I want to be able to offer these other Microsoft services that typically accessed via the Internet through my express route connection. But more typically what I'm going to do is private peering.



So private peering it's connecting an IP space I define for my on-prem or even maybe another cloud to the IP space of my virtual network. What I really want to think about is Azure just becomes an extension of my IP space. I don't want to think of it as these different thing, I want them to just have this free flowing IP space. And yes, site to site VPN does that, but companies don't like the fact that goes over the Internet even though it's encrypted, it is safe. Companies don't like it. And also for the variable latency. So what we have is this idea that there is this great big Microsoft WAN. It's backbone network and every Azure region is connected to it, redundant connections to it, so I can also think about this Region 2 for example. That's got redundant connections to this Microsoft backbone network as well, and they extend this backbone network to other locations. So one of the things you'll actually see is this idea of these carrier neutral facilities called peering points. Might also hear them called «Meet Mees». Remember you had your network. Well one of the things I can do is I can have a connection to the meet me. Which now connects your network to the Azure backbone. Maybe it's not this, maybe I actually have an MPLS and all my locations hang off the MPLS. Well now the MPLS connects to this meet me and in the middle here there's magic that does a cross connect between them.

So now again my network is connected and then what I do through ExpressRoute is from this gateway. So here. I have express route Private Peering. This IP space is now connected to this IP space. That's what private peering does. Have this private connection. I've connected this space together. Now this gateway here is an express route type gateway. I can have site to site VPN and express route in the same virtual network. Express route would get used first and then maybe VPN is a failover. Expressed route went down, it could fail over to the site to site VPN. I can even run site site VPN over the ExpressRoute which you might think why? Well remember site to site VPN is encrypted. ExpressRoute as a private connection. It is not an encrypted connection, it's private wire. Generally we don't worry about that stuff, but if you really were worried I could run the site site VPN over my private express route private peering to then encrypt the traffic inside. There is already a mechanism called Express Route macsec which encrypts, but it only encrypts at the meet me. So it's that space of air at the meet me between your router and the Microsoft Enterprise Edge routers that then go and connect. It's that because what's actually happening in here, at the meet me, is there's two Microsoft Edge routers that you get 2 connections. They run in active, active normally. There's maintenance, there's a failure and reduced to one. But normally I'm connected to two different Microsoft Enterprise Edge routers at this meet me that connect to my edge. Ordinarily, this connection is provided by some provider, Equinix, At&T etc... Microsoft don't provide the last mile. You work with some providers.

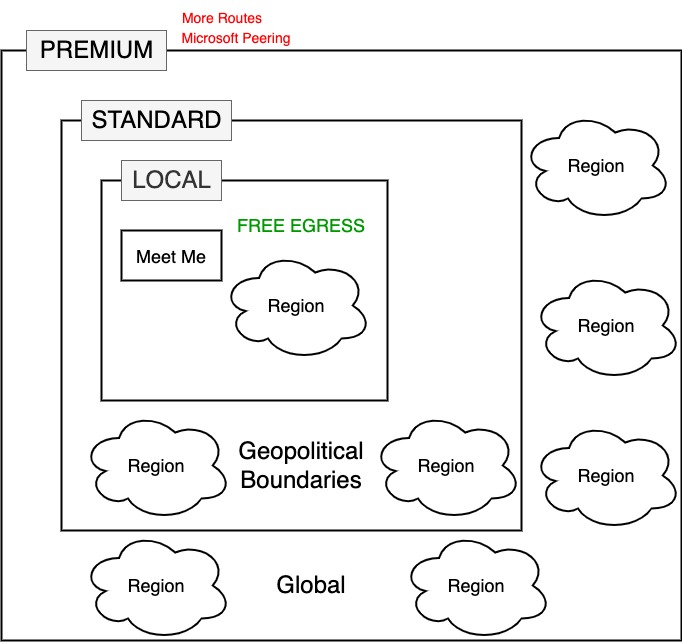
There is a so-called Express Route Direct. I'm a really big company and instead of relying on the provider for that last mile, I'm doing it myself. I'm dropping my own fiber, I'm dealing with levels of light. I get let with authorization from Microsoft. So at the meet me they connect my router into the Microsoft Enterprise Edge. And then on top of that, I create circuits of various sizes. So there are ways that I can do that, but generally that's really big companies. I could buy a 10 or 100 Gigabit per second ports and then fill it up with circuits on top of that. So I can absolutely do this as well.



Now realize from a resiliency perspective I might have multiple locations, so maybe this is, I don't know, SouthAsia and maybe this is Europe. So I might have another location over here. When it might have a connection. So this meet me, which primarily uses this region. But absolutely I could still have the ability to have connections so a single gateway can connect to multiple circuits. So I might have a different circuit over here, but this gateway could have a connection to both of them and maybe I've got some connection between them so I can still get to this location. My network here could get connected to both of those gateways because maybe they're needs VNets peered. Because I remember I don't want to use the ExpressRoute to talk between my virtual networks, so I would probably have peering. I could add certain resiliency by, this gateway actually connects to a circuit here and a circuit that lives there. So it's got different paths. And likewise, this would connect to both of them. So it has resiliency. So that same concept we saw with VPN. I would have here, this connects, this connects, that connects, that connects. And there's things I could do with preference. There's things I can do with proper path length. There's a whole set of hops that go on here to make it use the preferred one where it can.

The exact number of connections vary depending on my standard or my premium SKU. Premium add certain functionalities. For example, with standard I can connect to regions within the same geopolitical boundary. But if I do premium, I could have a circuit in the US but go and connect on a region in Europe for example. Premium also lets me do Microsoft peering, so there's other services. It also lets me advertise more routes over the network. It's more than standard.

One thing you have to consider is the traffic. You never ever pay for Ingress. So network traffic going into a region, so that could be coming from ExpressRoute, it could be coming from the Internet, you never pay for Ingress. Ingress. Is free. That could be ingress from the Internet, it could be ingress from on premises. Doesn't matter, you don't pay for ingress. Egress, so data leaving Azure. You, absolutely pay for it. You pay for data leaving Azure.



The exception to that is on this ExpressRoute. There's actually a service called Express Route Local. So, let me open the page on my browser. These are all the points where they're Peering points. And there's different this one, so there's different locations all around the world. We can see here lots and lots of them and they operate by different partners. Different partners can see that second column of the partners to operate that last mile and provide that connectivity. But you'll also see there's this 4th column local Azure region. So if you buy a local SKU. I don't pay egress but I can only talk to its local region for that meet me location. So that's the whole point. If I use express route local whatever region is the local region for that meet me location has the only region I can talk to. Now don't pay egress, so when the data is coming back to my on-prem I don't have to pay for that. But I can't.

Then there's the standard express route. So then it's the other regions in the same geopolitical boundary. So that's what express route standard would let me do. But the whole point of remember again if this one. And then finally, we have premium that is global. Any region all throughout the world. So there's different sets of boundaries I can think about. And obviously I pay more, I pay more for premium than I pay for standard, also get those additional routes plus, I can also do Microsoft peering. But there's additional capabilities. There's different types of express route depending on exactly what I need.

Now even for regular express route, there's metered and unmetered. Metered means I pay based on the amount of egress I do, which is what most people want. There is also an unmetered one, but it it's really expensive like so these are the types of gateway you can do and they talked about.

# Azure Virtual WAN

If I think about all the pictures that we saw, they were actually maybe a little bit scary. We saw hub with spokes, with gateways and maybe transit and maybe virtual appliances in there, it's like looks pretty horrible. I would never ever want to do that thing. Well, sometimes you want all of the control, you want all the responsibility, you want all the flexibility of what you can actually define. But if you don't, if I just want the service provided to me. Azure Virtual Wan, provides managed hubs per region. So I create an Azure virtual Wan in each region I want it, it creates a virtual hub. And then? It's different SKUs. There's a basic and a standard and really basic only support site to site VPN coming into that hub and then I connect VNets to it. That's it. With the standard it supports yes site to site VPN, but also ExpressRoute private peering. It enables the transit between the VNets, enables into hub connectivity. It has a whole bunch more stuff.