# AZ-104T00A – Administer Virtual Networking

Good day everyone. Before we start, any question regarding the topic of the past lesson? We discussed Azure Service Portfolio and Azure Resource Manager with tools and templates. I just want to share with you the results of the knowledge check of the past module. I’m happy to say that I saw a lot of right answers, here we have nine out of eleven, here ten out of eleven and here we have a perfect score. So, it means that the concepts explained have been understood. Of course, not 100% but It’s a good start to work on. Keep up the good work, please. If there are no questions, we can procced with the current module. Today’s topic is one of the pillars of the course. Virtual Networking. Modules four, five and six count one-third of the exam questions, so it’s very important to understand the networking topic in Azure. We are going to discuss Virtual Networks, Security Groups, Azure Firewall and at the end Azure DNS.

## Configure Virtual Networks

Azure virtual networks are the foundation upon which many or most other Azure resources are built. They allow resources like virtual machines to securely communicate with each other, with resources deployed to on-prem networks, and with the Internet. While a virtual network in Azure is quite similar to a physical network that you would see in on-prem datacenter, the virtual network offers additional benefits, including improved availability, scalability, and isolation.

Azure Networking Components

One of the major drivers for adopting a cloud solutions like Azure is to enable IT departments to transition server resources to the cloud. Moving resources to the cloud can save money and simplify administrative operations. Relocating resources removes the need to maintain expensive datacenters and for small and medium-sized companies, which might not have the expertise to maintain their own infrastructure, moving to the cloud is particularly appealing. Azure network services offer a range of components with functionalities and capabilities that can leverage this transition with reduced or no impacts.

Virtual Networks

This slide highlights the various subsystems of a virtual machine. Principally for our use, I'm looking at networking, so we have the virtual machine, and notice that it's not the VM resource as such that's placed in the VNet and on a subnet, but the network interface, which is a separate Azure resource that is then associated to that virtual machine. Long story short, it's the virtual NIC that houses or stores your TCP/IP configuration, not the virtual machine resource itself. Then we see, under the subnet box, a resource called network security group. The question I get asked most often on network security group is: “Is it a firewall?” The answer is no, it’s not a firewall even if they may seem to act like firewalls by handling inbound and outbound traffic. We will address Azure firewall later on. The network security group works at the transport layer, so, at layer four. Then, here we have a public IP. The VMs, by default, do not need a public IP address to get out to the internet. Rather, a public IP address allows inbound internet connections to that VM. And I can't overstate how important this is, don't put a public IP on the VM unless you have a defined business reason for doing so. If it's for management, there's so many other ways to do that. You can use Azure Bastion, you can use network address translation with Azure Firewall, Application Gateway, Azure Load Balancer. So many other ways to go. If you need to use a public IP address on a VM, consider using just‑in‑time VM access and Microsoft Defender for Cloud. Lastly, Azure has a separate DNS service where you can stand up both public, internet routable, and private, non‑internet accessible, DNS zones.

Plan Virtual Networks

There are four key terms or concepts that you must be familiar with before deploying an Azure virtual network. They include the address space, subnets, regions, and subscriptions. We already discussed regions and subscriptions so they should be clear concepts. Let’s start with address space. Whenever you deploy a virtual network in Azure, you’ll be prompted to define an address space. The address space that you define must consist of public or private addresses that conform to the RFC 1918. When you deploy resources and attach them to a virtual network in Azure, the private IP addresses that are assigned to them are pulled from the address space that you define. For example, if you were to provision a virtual network with an address space of 192.168.0.0/16, any resources that are connected to that virtual network would be assigned addresses from that range. Those devices would get addresses like 192.168.0.4, 192.168.0.5, and so on and so forth.

Create Subnets

Just as they do in a physical network, subnets allow you to segment a specific virtual network into one or more different subnetworks, hence the name subnets. When you do this, what you are really doing is carving out a portion of the virtual network's address space and assigning it to each subnet that you create. Azure resources can then be deployed to each subnet as necessary. Segmenting a virtual network address space into different subnets allows you to more efficiently leverage your addressing in a way that matches your organization’s requirements. Subnets also allow you to better secure resources through the use of network security groups. When you deploy a virtual network in Microsoft Azure, it is scoped to a single region or location. That being said, if you need to enable connectivity across different regions, you can connect multiple virtual networks from different regions through virtual network peering, which we will talk about in more detail in the next module. Just for your curiosity, when Microsoft completed the first 2 Azure datacenters in 2005/2006 one in a place called Quincy Washington, in the Eastern Washington, and the other in San Antonio Texas, they had to decide whether to lease the connection line or buy fiber optics to create a new private connection between those 2 datacenters and imagine what they chose? Mile after mile they have built the largest, or one of the largest private networks in the world, known as Microsoft Global Network, which has 175000 miles of private fiber optic cable that handles only the traffic between Azure Datacenters, that’s enough to go around the globe seven times. So, when we talk about interconnection between regions in Azure, we are not talking about internet network, we are talking about high-speed private network. In addition to being scoped to a single region or location, each virtual network that you deploy is scoped to a specific subscription.

There are several best practices that you should keep in mind as you build out your virtual networks in Microsoft Azure. First, and foremost, when deploying a virtual network, you need to ensure that the address space that you define for that virtual network does not overlap with any other network ranges that your organization uses. A real-world example of this best practice would be a situation where you plan to deploy a virtual network that will be connected to your physical on-prem network over a site-to-site VPN. If your on-prem network consists of a 192.168.0.0/16 address space or subnet, you wouldn’t want your Azure virtual network to overlap that specific address space or subnet because such an overlap could cause routing issues. Speaking of subnets, you should never create a subnet that encompasses the entire address space of the virtual network. In other words, if you deploy a virtual network with an address space of 192.168.0.0/16, you really should not be creating a subnet for that virtual network with the same address range. Instead, you should be planning ahead and leaving some of the address space available for future use. This means that any subnets that are defined on that virtual network should only use a portion of the complete virtual network address space. It’s also recommended that instead of defining many smaller virtual networks, you should define fewer larger virtual networks. By leveraging fewer large virtual networks rather than a bunch of little virtual networks, you can prevent or minimize management overhead. And last but not least, you should be securing all virtual networks that you create with network security groups. By assigning network security groups to the subnets that have been defined for a virtual network, you can retain control over what traffic passes between those subnets.

Create Virtual Networks

Since last time we talked about Templates, let’s create 2 virtual machines inside a new VNet through a Template. Let’s have a look at the 2 files. First, we create a new resource group with powershell with the command

**$rgName = “Demo-VNet”**

**$location = “Switzerland North“**

**$tags = @{“Environment”=”Demo-VNet”}**

**New-AzResourceGroup -Name $rgName -location $location -tag $tags**

Then we create the virtual machines with the command:

**New-AzResourceGroupDeployment -ResourceGroup $rgName -TemplateFile demo-vnet-template.json -TemplateParameterFile demo-template-parameters.json**

We have these 2 virtual machines inside the same VNet but in 2 different subnets. Now let’s assign an IP address (just for testing) to the first one to be able to connect through RDP and test the connectivity between the 2 VMs. We will implement the connectivity through Bastion host in a second stage. Since by default ICMP traffic is disabled, we have to enable through PS command ***New-NetFirewallRule -DisplayName “Allow ICMPv4-in” -Protocol ICMPv4.*** But we can connect through RDP.

On-premises and Azure IP Addressing

On the left you can see a typical on-premises network design (apologize to my colleagues in the network team, I know it’s not so simple, but just to have something to discuss on). This diagram shows a very simplified version of a typical on-premises network. On the routers facing the internet service provider (ISP), you have public IP addresses that are used by your outbound internet traffic as their source. These addresses also are used for inbound traffic across the internet. The ISP might issue you a block of IP addresses to assign to your devices, or you might have your own block of public IP addresses that your organization owns and controls. You can assign these addresses to systems that you would like to make accessible from the internet, such as web servers. The perimeter network and internal zone have private IP addresses. In the perimeter network and internal zone, the IP addresses that are assigned to these devices aren't accessible over the internet. The administrator has full control over the IP address assignment, name resolution, security settings, and security rules. There are three ranges of non-routable IP addresses that are designed for internal networks that won't be sent over internet routers: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16. The administrator can add or remove on-premises subnets to accommodate network devices and services. The number of subnets and IP addresses you can have in your on-premises network depends on the Classless Inter-Domain Routing (CIDR) for the IP address block. On the right side another simplified version of an Azure Virtual network design. In Azure, the network design has features and functions that are similar to an on-premises network but the network structure is different. The Azure network does not follow the typical on-premises hierarchical network design. The Azure network provides the ability to scale up and scale down infrastructure based on demand. Provisioning in the Azure network happens in a matter of seconds. There are no hardware devices, like routers or switches. The entire infrastructure is virtual, and you can slice it into chunks that suit your requirements. In Azure, you typically would implement a network security group and a firewall. You'd use subnets to isolate front-end services, including web servers and DNS, and back-end services like databases and storage systems. Network security groups filter internal and external traffic at the network layer. A firewall has more extensive capabilities for network layer filtering and application layer filtering. By deploying both network security groups and a firewall, you get improved isolation of resources for a secure network architecture. In Azure there are IP Ranges reserved by Azure Platform: 169.254.0.0/16 for Link-local and most important 168.63.129.16/32. This is the so-called Azure Wire Server service. The Wire Server is responsible for DHCP. This allows your resources placed on those subnets in your VNet to receive private, non‑routable IP addresses automatically. The Wire Server also is the default gateway, and unless you take actions to restrict outbound internet connections, your virtual machines can in fact get out onto the internet, and they'll use the Wire Server as its default gateway. Wire Server also takes care of Azure‑provided DNS, hostname resolution. And lastly, the Wire Server is responsible for VM Agent communication. In other words, heartbeat, messages, status between your virtual machines and the Azure backplane.

## Plan IP Addressing

A good Azure IP addressing schema provides flexibility, room for growth, and integration with on-premises networks. The schema ensures that communication works for deployed resources, minimizes public exposure of systems, and gives the organization flexibility in its network. If not properly designed, systems might not be able to communicate, and additional work will be required to remediate. To integrate resources in an Azure virtual network with resources in your on-premises network, you must understand how you can connect those resources and how to configure IP addresses. You can assign IP addresses to Azure resources to communicate with other Azure resources, your on-premises network, and the internet. There are two types of Azure IP addresses: *private* and *public*.

**Private IP addresses** enable communication within an Azure virtual network and your on-premises network. You create a private IP address for your resource when you use a VPN gateway or Azure ExpressRoute circuit to extend your network to Azure.

**Public IP addresses** allow your resource to communicate with the internet. You can create a public IP address to connect with Azure public-facing services.

IP addresses can be statically assigned or dynamically assigned, you can separate dynamically and statically assigned IP resources into different subnets. Static IP address are best for certain situations, such as: 1) DNS name resolution, where a change in the IP address requires updating host records. 2) IP-address based security models that require apps or services to have a static IP address. 3) TLS/SSL certificates linked to an IP address. 4) Firewall rules that allow or deny traffic by using IP address range. 5) Role-based virtual machines such as Domain Controllers and DNS servers.

Create Public IP Address

DEMO with Azure Portal. Create one Public IP Address. IP version IPv4 or IPv6 or both. SKU Standard or Basic (depend on the SKU of the Load Balancer associated) and with Basic we can choose the IP address assignment model (static or dynamic).

**Dynamic** addresses are assigned only after a public IP address is associated to an Azure resource, and the resource is started for the first time. Dynamic addresses can change if they're assigned to a resource, such as a virtual machine. The virtual machine must be stopped (deallocated), and then restarted. The address remains the same if a virtual machine is rebooted or stopped (but not deallocated). Dynamic addresses are released when a public IP address resource is dissociated from a resource.

**Static** addresses are assigned when a public IP address is created. Static addresses aren't released until a public IP address resource is deleted. If the address isn't associated to a resource, you can change the assignment method after the address is created. If the address is associated to a resource, you might not be able to change the assignment method. If you select **IPv6** for the IP version, the assignment method must be **Dynamic** for the Basic SKU. Standard SKU addresses are **Static** for both IPv4 and IPv6 addresses.

Associate public IP addresses

A public IP address resource can be associated with virtual machine network interfaces, internet-facing load balancers, VPN gateways, and application gateways. You can associate your resource with both dynamic and static public IP addresses

Associate private IP addresses

A private IP address resource can be associated with virtual machine network interfaces, internal load balancers, and application gateways. Azure can provide an IP address (dynamic assignment) or you can assign the IP address (static assignment).

Demonstration – Virtual Networks Powershell

Create a virtual network using PowerShell.

*$myVNet02 = New-AzVirtualNetwork -ResourceGroupName RG-Network -Location switzerlandnorth -Name myVNet02 -AddressPrefix 10.0.0.0/16*

*Get-AzVirtualNetwork -Name myVNet02*

*$mySubnet02 = Add-AzVirtualNetworkSubnetConfig -Name mySubnet02 -AddressPrefix 10.0.0.0/24 -VirtualNetwork $myVNet02*

*Get-AzVirtualNetworkSubnetConfig -Name mySubnet02 -VirtualNetwork $myVNet02*

*$mySubnet02 | Set-AzVirtualNetwork*

## Configure Network Security Groups

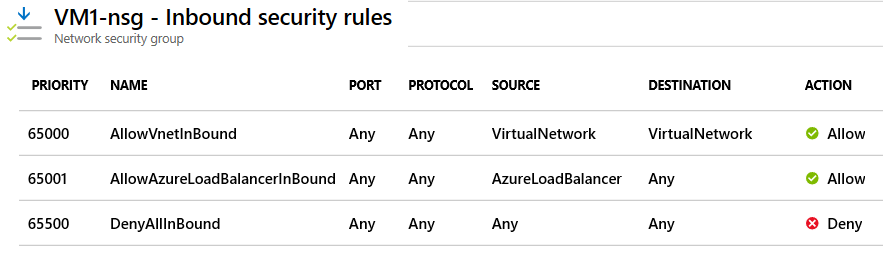
Network security groups are used to filter network traffic to and from [Azure](https://cloudacademy.com/library/azure/) resources in an [Azure virtual network](https://cloudacademy.com/course/implementing-azure-network-security/azure-virtual-networks/). When you create a network security group, that group will contain security rules that allow or deny inbound network traffic to or outbound network traffic from many types of Azure resources

Implement Network Security Groups (NSGs)

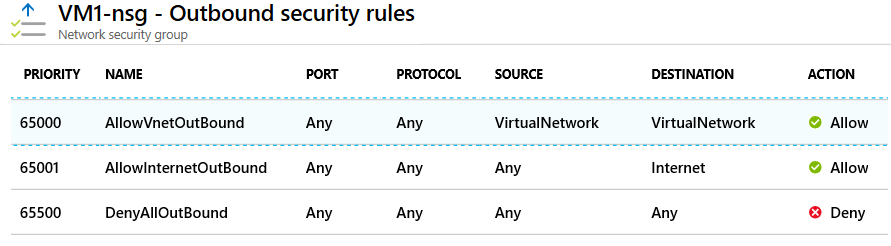
A network security group can be associated to a subnet or a network interface and can be associated multiple times. Let’s create a new security group in the Azure Portal. A network security group can contain many security rules as you require or it can contain zero security rules.

Determine NSG Rules

Each rule defined within a network security group requires a unique name within the security group, a priority, a source or destination, a protocol, a direction, a port range, and an action. The priority is a number between 100 and 4,096. This priority determines the order in which rules are processed. Rules with lower priority numbers are processed before those with higher numbers and once traffic matches a specific rule, processing of remaining rules stops. When you define a source or destination for a security rule, what you are doing is telling the rule whether the destination is an individual IP address, any IP address, a CIDR block, a particular service tag, or an application security group. The protocols that you can configure within a security rule include TCP, UDP, ICMP, or Any, and as far as directions go, you can configure security rules to apply to either inbound or outbound traffic. When defining a security rule, you can specify an individual port for it to apply to or you can specify a range of ports. For example, you can specify something like port 443 or you can specify a range like ports 10,000 to 11,000. Defining ranges allows you to create fewer security rules, since you don't have to create an individual rule for each port. The action that you define for a security rule can be either allow or deny. This determines what happens when the rule is matched. Azure creates several default security rules within each network security group, including inbound traffic and outbound traffic. Examples of default rules include DenyAllInbound traffic and AllowInternetOutbound traffic. Azure creates the default security rules in each network security group that you create and you can’t remove them. You can override a default security rule by creating another security rule that has a higher Priority setting for your network security group. These are the default rules. These rules deny all inbound traffic except traffic from your virtual network and Azure Load Balancers:



The default outbound security rules only allow outbound traffic to the internet and your virtual network.



Determine NSG Effective Rules

Each network security group and its defined security rules are evaluated independently. Azure processes the conditions in each rule defined for each virtual machine in your configuration. For inbound traffic, Azure first processes network security group security rules for any associated subnets and then any associated network interfaces. For outbound traffic, the process is reversed. For both inbound and outbound evaluation process, Azure also checks how to apply the rules for intra-subnet traffic. Let’s have a look at this scenario that shows network security groups (NSGs) controlling traffic to virtual machines (VMs). The configuration requires security rules to manage network traffic to and from the internet over TCP port 80 via the network interface. In this virtual network configuration, there are three subnets. Subnet 1 contains two virtual machines: VM 1 and VM 2. Subnet 2 and Subnet 3 each contain one virtual machine: VM 3 and VM 4, respectively. Each VM has a network interface card (NIC). Azure evaluates each NSG configuration to determine the effective security rules as you can see in the slide. For the inbound traffic, Azure identifies if the VMs are members of an NSG, and if they have an associated subnet or NIC. When an NSG is created, Azure creates the default security rule DenyAllInbound for the group. The default behavior is to deny all inbound traffic from the internet. If an NSG has a subnet or NIC, the rules for the subnet or NIC can override the default Azure security rules and NSG inbound rules for a subnet in a VM take precedence over NSG inbound rules for a NIC in the same VM. For the outbound traffic, Azure processes rules for outbound traffic by first examining NSG associations for NICs in all VMs. When an NSG is created, Azure creates the default security rule AllowInternetOutbound for the group. The default behavior is to allow all outbound traffic to the internet. If an NSG has a subnet or NIC, the rules for the subnet or NIC can override the default Azure security rules and NSG outbound rules for a NIC in a VM take precedence over NSG outbound rules for a subnet in the same VM. If you have several network security groups and aren't sure which security rules are being applied, you can use the **Effective security rules** link in the Azure portal. You can use the link to verify which security rules are applied to your machines, subnets, and network interfaces.

Create NSG rules

Let me show you directly from the portal, I just wanted to quickly run through an overview of the different options that are available when configuring a rule. As we saw before, when creating an NSG, Azure creates default inbound and outbound rules. If I want to create other inbound rules, I need to click over here on the **Inbound Security rules** option. Of course, I would click **Outbound Security Rules** to create an outbound rule. Clicking on Add opens up the new rule blade. Now, in this source box here I have a few different options. I can set my source to Any, Application security group, IP addresses, or Service tag. If I select Application security group here, I'm prompted to select one or more existing application security groups. If I select IP addresses, I can specify either source IP addresses, or entire CIDR ranges. I can specify a single value, or a comma-separated list of multiple values. For example, I could specify multiple values like 10.0.0.0/16 and then separate it with a comma and then an individual address, 192.168.0.0. Now, I should point out that if I specify an IP address of an Azure VM here, I need to ensure that I specify the private IP address and not the public IP address that's assigned to the VM. This is because security rules are processed after Azure translates the public IP to a private IP for inbound security rules and before [Azure](https://cloudacademy.com/library/azure/) translates a private IP to a public IP for outbound rules. Now, if I select Service tag here, I can then select one service tag. These service tags are predefined identifiers for different categories of IP addresses. They include things like HTTP, RDP, storage, load balancers, you name it. There are really too many options to list here, so I do suggest that you play around with this to see all the different options. Now, for the source port ranges field here, I can specify a single port like port 80. Now, I can also specify a range of ports like 1024, for example, through 1200. I can even specify a comma-delimited list of port ranges, or even ports. For example, I can do port 80,443,3389 and then I can even do like 1000 to 1200. So, I can use comma-delimited lists as well. Now what I can also do, and if you see down here at the bottom, the recommended value. The recommended value for source port ranges is \* or Any, and this is because port filtering, as Microsoft notes here, is typically used with a destination port. So, we can leave this at \* for Any as well. The destination field offers me some additional options here. I can select Any, IP Addresses, Virtual Network, or Application security group. As was the case up here with source, if I select Application security group, it's going to ask me to specify a defined security group. I don't have any configured here, so it's telling me as much right here. And then of course, if I specify an IP address here, the same rules apply in my destination IP addresses CIDR ranges as applied up here in my source IP addresses and CIDR ranges. As was the case in the source, if the IP address that I specify is assigned to an Azure VM, I need to make sure that I specify the private IP, not the public IP, assigned to that VM for the same reasons I mentioned up here. Now, selecting virtual network here for destination, which is actually a service tag. What this means is that traffic is allowed to all IP addresses within the address space of the specified virtual network. Now, selecting virtual network here, which is actually a service tag, means that traffic is allowed to all IP addresses within the address space of the virtual network. The destination port ranges option allows me to specify a single value or a comma-separated list of values that specify the destination ports that I'm allowing traffic to. The protocol option here allows me to select Any protocol, TCP, UDP, or ICMP. So, I can filter down on what protocols I'm allowing. And then, of course, the action field determines the result of matching the rule. I can choose to either allow or deny. Priority allows me to provide a value between 100 and 4,096. This has to be unique for all security rules within the network security group. Security rules are processed in priority order and what that means is, the lower the number, the higher the priority. Microsoft actually recommends that you leave a gap between priority numbers when you create your rules. So, for example, instead of setting priorities of one, two, and three, use values like 100, 200, and 300. Leaving gaps makes it easier to add rules in the future without juggling priority values. Then when you create a rule, the name for the rule needs to be unique within the network security group. It can be up to 80 characters in length and it must begin with a letter, or a number, end with a letter, number, or underscore, and it can only contain letters, numbers, underscores, periods, or hyphens. Essentially no spaces. And, of course, the description here is optional. We don't have the little red star here telling us it's mandatory. Let’s create an inbound rule that allows port 80 to all a subnet.

Implement Application Security Groups

During the demo we saw an option regarding Application Security Group. So, what exactly are application security groups? An application security group is a grouping of virtual network interfaces that is used to configure network security for the virtual machines that the NICs are attached to. Using an application security group allows you to define network security policies based on the group that you define. You can even reuse your defined security policy at scale without the need to manually maintain a list of included IP addresses. While application security groups offer a more streamlined way to secure applications running on VMs, they do come with some constraints. For example, you are limited to 30,000 application security groups that can be defined within a subscription. In addition, you can specify only one application security group as the source and destination when configuring a security rule. You are not allowed to specify multiple application security groups in either the source or destination. Also, when configuring an application security group, all network interfaces that are assigned to the group must exist within the same [virtual network](https://cloudacademy.com/course/implementing-azure-network-security/azure-virtual-networks/) as the first network interface that is assigned to the application security group. For example, if you create an application security group and the first NIC that you assign to is connected to vNet1, all subsequent NICs assigned to the group must also be connected to vNet1. You can't add NICs from different virtual networks to the same application security group. Lastly, if you specify an application security group as the source and destination in a security rule, the NICs in both groups must exist within the same virtual network. So, for example, if you define a group called Group1 that contains NICs from VNet1 and then, you define another group called Group2 that contains NICs from VNet2, you can't assign Group1 as the source and Group2 as the destination in a rule that you define because all NICs for both the source and destination application security groups need to exist in the same virtual network.

Demonstration - Network Security Groups and Application Security Groups

## Configure Azure Firewall

Azure Firewall is a cloud-based network security service that is designed to protect [Azure Virtual Network](https://cloudacademy.com/course/implementing-azure-network-security/azure-virtual-networks/) resources. Because this is a managed service, its added advantages are that it offers high availability and scalability.

Determine Azure Firewall Uses

It's fully stateful and features built-in high availability as well as unrestricted cloud scalability. Using Azure Firewall, organizations can centrally create application and network connectivity policies across subscriptions and networks. Because Azure Firewall uses a static, public IP address for the virtual network resources behind it, it permits outside firewalls to identify traffic that comes from the protected virtual network. Because Azure Firewall is fully integrated with [Azure](https://cloudacademy.com/library/azure/) Monitor, organizations can also perform logging and analysis against it. Azure firewall offers several key features, including built-in high availability, availability-zone spanning, unrestricted cloud scalability, application FQDN traffic-filtering rules, and network traffic-filtering rules. Other features of Azure Firewall include things like: support for FQDN tags, service tags, and threat intelligence. You can even associate up to 100 different public IP addresses with an Azure Firewall.

Create Azure Firewalls

When you deploy a firewall, the recommended approach is to implement a hub-spoke network topology. The **hub** is a virtual network in Azure that acts as a central point of connectivity to your on-premises network. **Spokes** are virtual networks that peer with the hub, and can be used to isolate workloads. Traffic flows between an on-premises datacenter and the hub network through an Azure connection, such as Azure ExpressRoute, Azure VPN Gateway, or Azure Bastion. It helps to support workloads in different environments that require shared services, but not connectivity to each other. Examples of these environments include development and testing where each requires DNS for example. Place shared services in the hub virtual network, and deploy each environment to a spoke to maintain isolation.

Create Azure Firewall Rules

Coming back to Azure Firewall. By default, it denies all traffic through your virtual network. The purpose of the default behaviour is to provide the highest level of protection against malicious or unknown access. To allow traffic for a particular resource or service, you need to define rules to control the specific traffic. There are three kinds of rules you can configure for Azure Firewall: NAT, network, and application. The rules are defined in the Azure portal. When a packet arrives on a designated port in your network, the packet is inspected to determine if it's allowed. Azure Firewall processes the packet by evaluating it against your rules in the following order: Network rules then Application rules. If a rule is found that allows the packet through, no remaining network or application rules are checked against the packet. After a packet is allowed, Azure Firewall checks for NAT rules that define how to route the traffic.

DEMO Azure Firewall

To prepare for this demonstration, I've already set up a virtual network and a few subnets that we're going to use. This vNet-FirewallDemo virtual network is the virtual network we're going to work with. Within this virtual network, I've created two subnets. I did this to simulate a production environment.

The first subnet is called subnet-workloads. The second subnet is called subnet-jump. In the workloads subnet, I have a VM called PROD01, this PROD01 can simulate an app server or a file server or whatever the case might be. Now I've configured PROD01 without a public IP and I've configured the public inbound ports for it to none.

In my jump subnet, I also have a VM running, this VM is called JUMP01. And it's going to serve as my jump box to access my PROD01 in the workloads subnet. The address range for the subnet workloads subnet is 10.0.2.0/24. The address range for subnet-jump is 10.0.3.0/24.

So with our lab environment explained here, let's get into the deployment of Azure firewall.

To start our configuration, we need to create a subnet with a specific name. We need to create a subnet called AzureFirewallSubnet within our [virtual network](https://cloudacademy.com/course/implementing-azure-network-security/azure-virtual-networks/) here, this subnet will house our firewall.

So we'll go ahead and create my sub net and I'm not going to do a play-by-play here because I assume you already know how to do this. What I'm going to do here is give this a range of 10.0.1.0/26. And this is because Microsoft recommends a size of /26 for the firewall subnet.

Okay, so we have our firewall subnet here with our range of 10.0.1.0/26. With our firewall subnet created, we can deploy our firewall into the vNet-FirewallDemo virtual network. To do this, I'm going to click Create a Resource from my portal page here and I'm just going to search for firewall and then we'll select it from the list. We'll go ahead and create.

Now, from this Create page, I'm going to configure my new firewall that we're deploying here. I need to make sure that I'm deploying into my intended subscription and into the resource group I intend. So we'll go ahead and select our DemoRG. I'm going to call my firewall, simply MyFirewall. And then I'll set my region to the same as my other resources, which is Switzerland North, and I'm not worried about high availability here. So I'll leave the availability zone set to none for this exercise here.

Now, in this, choose a virtual network here. What I need to do is either create a new virtual network for this firewall or use an existing one. So we're going to use an existing one. We'll select our vNet-FirewallDemo and then what I need to do is specify a public IP address. Now, I can either create a new one or select an existing one. So bear with me here while I create a new one. Just going to call it PublicIP-FW.

Now, I should point out here that the SKU section here for my public IP needs to be standard and it needs to be static. And we can see here that these are already set for us. So we'll okay this and now what I'll do is click Review and Create here to create the firewall. And then we'll go ahead and create it. And this deployment can take a few minutes to finish. So, all clear so far? Do you have questions?

So, with the deployment complete, I can click on Go to Resource here to see my firewall details. Now I need to note the private IP here, because I'm going to use this later on when I create my default route. Now my default route is going to send all traffic from the workload subnet through the firewall. To create my default route, what I'm going to do is browse to All Services here. And then what I'm going to do is browse to Networking and then I'm going to choose Route tables. And then I can either click Create Route Table or Add and we're going to create a new route table and we'll call it MyRouteTable.

We're going to deploy into my lab subscription and into my demo resource group. Location again will be Switzerland North, and if we planned to associate this route table to a subnet in a virtual network that's connected to an on-prem network through a VPN gateway, disabling propagation would cause our on-prem routes to not propagate to the network interfaces in our subnet. We're going to leave propagation enabled here. And then with all my choices made, I can click Create.

Now once it's been deployed, and it takes a few moments, I can refresh here and see my new route table. Let's go ahead and refresh here. And we can see my route table's now showing up. So now what I need to do is associate my table with my virtual network. So I'll click my route table here, and then select subnets. We can see there are currently no subnets associated with the route table, so we'll click Associate here. And then from here, I will select my vNet-FirewallDemo virtual network. Now the subnet that I choose here is going to be my workload subnet. Then we'll Okay it.

Next, I need to click on Routes, and then click Add. This is where I'm going to define my actual route that sends all traffic through the firewall. I'm going to call my route DefaultRoute. For address prefix, I need to type 0.0.0.0/0, since I'm sending all traffic through this route. The next hop type here should be set to virtual appliance. And this is because my next hop is Azure firewall. Now you might be saying to yourself, I thought Azure firewall was a service, it is but for this exercise, it's seen as a virtual appliance.

Now for the next hop address, I need to provide the private IP address for my firewall. Remember, we noted this earlier, At this point, I can click Okay. And this adds my route to my route table.

So now that we have our default route configured, what we're going to do is create an application rule that allows outbound access to www.avaloq.com. Essentially, what we're doing is locking everything down to just avaloq.com. To do this, we're going to browse to our firewall. Let's go back out, to home, go to our firewall here. And then what we're going to do is go to Rules under Settings here. And since we're setting up an application rule, we'll choose the Application Rule Collection tab. And then under the Application Rule Collection tab we'll click Add Application Rule Collection.

We're going to call our collection here, MyAppCollection. And what we'll do here, is give it a priority of 200. Since we want to allow traffic, we're going to select Allow here for our action. And now under rules, we want to target FQDNs. So what we'll do is we'll provide a name for our new rule, and we'll call it AllowAvaloq.

For source addresses, we need to specify what network is allowed to reach Avaloq site. So for this exercise, we're going to allow the workload subnet, the 10.0.2.0/24. And for protocols, we'll allow HTTP and HTTPS. And then of course, for the target FQDNs, we're going to target www.avaloq.com. And then what I'll do, is I'll add my rule here.

Okay, so now I've shown you how to create a route table, a default route, and an application rule. Now what I'm going to do is show you how to create a network rule. The network rule we're going to create allows outbound access to two different IP addresses on port 53, which is DNS. Because we're going to need DNS to resolve the avaloq.com domain name.

To create our network rule, we need to select the Network Rule Collection tab. Now from here, we'll choose the option to add a network rule collection and we'll call this NetworkCollection. Again, we'll set our priority to 200 and we're going to allow our traffic. At this point, we need to define our rule. So under IP addresses, under the Rule section here. For our name, we're going to call it AllowDNS. We'll choose UDP for the protocol since DNS is UDP traffic. For this source address here, we're going to set the source to our workload subnet which is 10.0.2.0/24. And then we're going to specify Google's public DNS servers of 8.8.8.8 and 8.8.4.4 for our destinations and then we can use a comma here. And for destination ports here, we'll use Port 53, because that's what DNS operates over. Then we'll go ahead and click Add.

So what we just did was create a rule that allows DNS traffic through our firewall, but only to Google's public DNS servers. And we'll let this rule appear here, and then we'll move on to the next step. Okay, so now our new rule is in place. So to test things out, we need to configure the primary and secondary DNS servers that our virtual machine is going to use. Now, just to be clear, this isn't an Azure firewall requirement, we're just doing this because it's a test environment and we're testing things out here in a demo.

So what I'm going to do is browse to the network interface for our PROD01 virtual machine. So, let's go back here into this network interface here, the PROD01215 network interface here is the NIC for PROD01. So we'll select our NIC here. And what we're going to do is change the DNS servers to custom and specify Google's servers here. And we'll save it.

Now once this updates, what I'll do is I'll go out to my PROD01 VM here and we'll start him up. We'll let this VM come up and then what we'll do is we'll test our rules to make sure everything's working the way it's supposed to.

So our PROD01 is running, but we can see we have no public IP address here. So we can't RDP to it directly and this was by design. So what we're going to do is go out to our JUMP01 box here that I deployed. And he is in the process of starting up, we're going to connect to our JUMP01. And then from there, jump over to PROD01. So we'll go ahead and connect here, now that it's running. And I'm just connecting off-screen here. And I'll drag this into my live screen.

Now I'll minimize this and remember, we're going to connect to PROD01 and that's 10.0.2.4. All right, so JUMP01 is up here. Let's launch our RDP console here or RDP client, I should say. And these are some slower lab machines so bear with me here. So the first test we're going to do here is see if we can actually RDP to our PROD server from JUMP. And it does appear that it's going to let me so now that I'm on my PROD01 VM, I can launch a browser here and try to browse to avaloq.com. What we'll do here is we'll minimize and open Chrome here. And what we'll do is we'll browse to www.avaloq.com. And you can see we've actually hit the page.

Now, the reason we have a mess here with no styling is because our firewall is blocking all traffic to everything but avaloq.com. So a lot of the styling is coming from an outside site or like a CDN. Now, if I try to go to, for example, microsoft.com, I can't even get to the site and it tells me that the request is denied because there's no rule matched. So, we can see that I was allowed to hit Avaloq but wasn't allowed to hit microsoft.com, I'm blocked by the firewall. This confirms that my firewall rules are working. It confirms that I can only browse to the allowed FQDN and it confirms that DNS access to Google's DNS servers is working as well. Otherwise, I wouldn't have been able to resolve the names. So with that, you now know how to create a route table, a default route application rules, and network rules. You now know how to create and configure an Azure firewall.