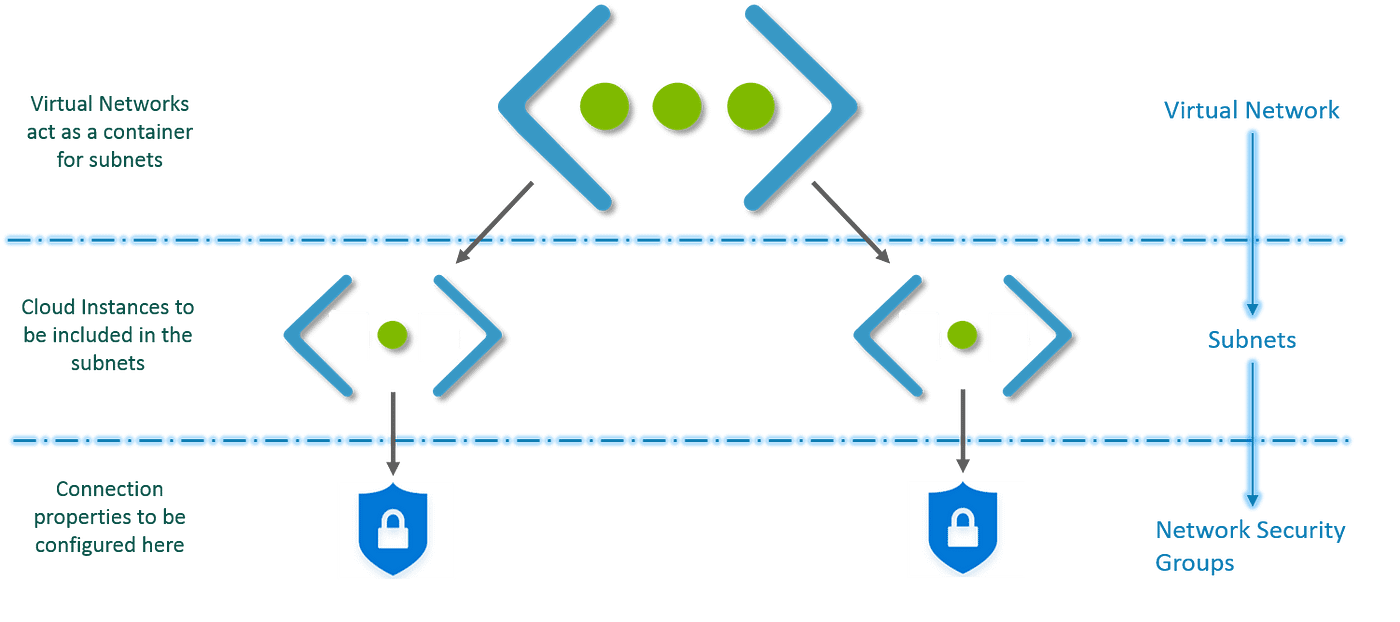
**Cloud Networking**

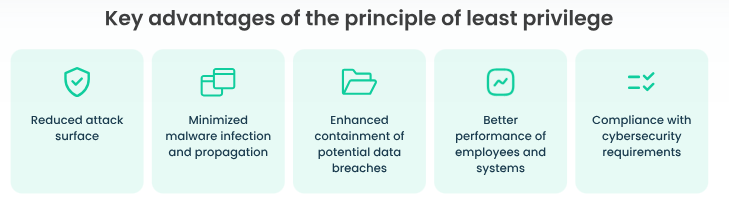
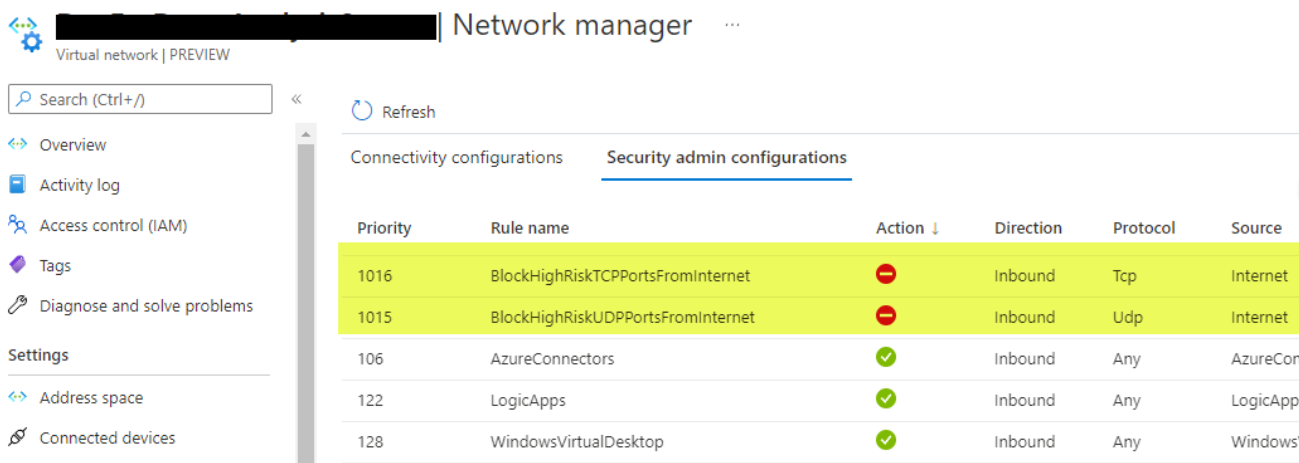
**Azure Virtual Networks: Building Secure and Scalable Architectures**

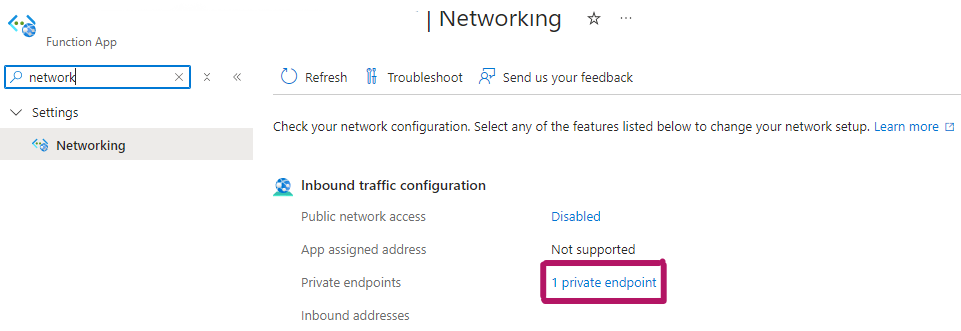
****

The foundation of any robust cloud environment is a well-designed virtual network. In Azure, a Virtual Network (VNet) acts as a private network within the public cloud, allowing you to securely deploy and manage your virtual machines (VMs), applications, and other resources. It provides isolation from the public internet and enables controlled communication between your resources.

**Security Best Practices - Azure**

Here are some key security best practices to consider when configuring subnets within your Azure VNet:

* **Principle of Least Privilege:** Grant the minimum permissions and access required for resources within a subnet. This minimizes potential damage in case of security breaches.
* **Just-in-Time (JIT) Access:** Implement JIT access for resources instead of providing constant unrestricted access. This reduces the attack surface.
* **Deny-All Default Rule:** Utilize a default "Deny All" inbound rule within your NSG. Subsequently, explicitly allow only the necessary traffic through specific rules. This approach provides a more secure baseline.
* **Restrict Source IP Addresses:** Don't allow inbound traffic from the internet (\*) for all resources. Specify the allowed source IP addresses or ranges for enhanced security.



* **Monitor and Audit:** Regularly monitor and audit your NSG rules and network activity to identify suspicious activity and potential security vulnerabilities.
* **Keep Software Updated:** Ensure all VMs and resources within your subnets have the latest security patches and updates applied to prevent known vulnerabilities.

**Building Blocks of a Secure Azure Network Architecture**

Imagine your organization's Azure environment as a well-organized city. Like physical networks, a secure Azure network architecture often utilizes multiple subnets within a VNet, each serving a specific purpose.

For instance **Application Subnet** (Houses your application VMs, ensuring secure communication within the controlled environment of the VNet), **Database Subnet (**Isolates your database VMs, providing an extra layer of security for your critical data), M**anaged Instances Subnet (**Dedicated subnet for Azure Managed Instances, offering additional security and performance benefits), **PaaS Services Subnet** (Groups Azure PaaS services like Azure SQL Database or Azure App Service, allowing controlled access to resources) etc.

By utilizing multiple subnets, you can segment your network logically, apply granular security policies, and optimize resource utilization.

**Step-by-Step Guide to Configuring Subnets in Azure**

Now, let's dive into the practical steps of creating a new subnet within your VNet:

**Step 1: Subnet - Granting Virtual Space.**

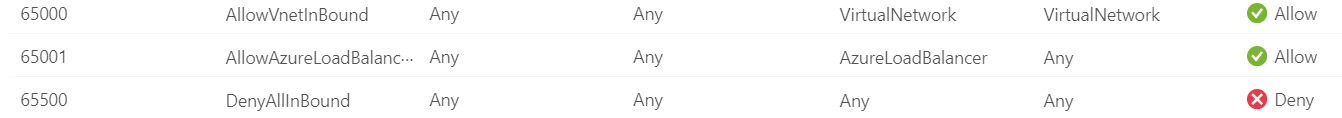
*Network within a Network, logical subdivision of an IP network.*

1. **Locate your VNet:** Navigate to the desired virtual network within the Azure portal.
2. **Add a Subnet:** Click on "Subnets" and then the "+" sign to initiate subnet creation.
3. **Descriptive Naming:** Assign a meaningful/complaint name to your subnet as per you organization’s policy, reflecting its purpose.
4. **IP Address Range:** Define a unique IP address range for the subnet, ensuring it doesn't overlap with existing ones within the VNet.

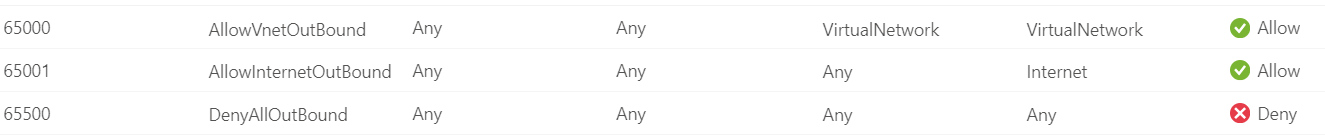
**Step 2: Network Security Groups (NSGs) - Securing the Subnet**

*Imagine an NSG as a digital bouncer guarding the entrance to your subnet. It controls inbound and outbound network traffic, ensuring only authorized communication flows through. Here's how to configure an NSG:*

1. **Craft an NSG:** Create a new Network Security Group with a suitable/compliant name as per organization’s policy .
2. **Link the NSG:** Attach this NSG to your newly created subnet.
3. **Inbound Security Rules:** Define rules specifying allowed traffic. Click "+" to configure details:
   * **Source:** Specify IP addresses or ranges permitted access (initially, you might use "\*" for broader access).
   * **Destination Port:** Define the port(s) your applications require (e.g., 80 for HTTP).
   * **Protocol:** Choose TCP, UDP, or the appropriate protocol for your traffic.
   * **Action:** Set this to "Allow" for desired traffic and "Deny" to block it.

Inbound

Outbound:



|  |  |
| --- | --- |
| **Property** | **Value** |
| Resource Group Name | MyResourceGroup |
| Network Security Group Name | MyNsg |
| Rule Name | Allow\_HTTP |
| Priority | 100 |
| Source Address Prefixes | \* (Any) |
| Source Port Ranges | \* (Any) |
| Destination Address Prefixes | \* (Any) |
| Destination Port Ranges | 80 (HTTP) |
| Access | Allow |
| Protocol | TCP |
| Direction | Inbound |
| Description | Allow HTTP traffic |

**Step 3: Route Table - Charting the Course for Traffic Flow**

Think of a Route Table as a map for your subnet's traffic. It guides data packets to their intended destinations. Here's how to configure one:

1. **Create a Route Table:** Craft a new Route Table with a descriptive name (e.g., **DevApps-Routing**).
2. **Connect the Route Table:** Link this Route Table to your subnet.
3. **Define Routes:** Within the table, specify routes for traffic flow:
   * **Destination:** Specify the IP address range your traffic needs to reach (e.g., the internet).
   * **Next Hop:** Choose "Internet" for web traffic or "Virtual Appliance" if routing through a specific device.
   * **Next Hop IP:** If using a virtual appliance, provide its IP address.

|  |  |
| --- | --- |
| **Property** | **Value** |
| Resource Group Name | MyResourceGroup |
| Route Table Name | MyRouteTable |
| Subnet Name | MySubnet |
| Route Name | RouteToVirtualNetwork |
| Address Prefix | VirtualNetworkAddressPrefix |
| Next Hop Type | VirtualAppliance (or Internet, VirtualNetworkGateway, VnetLocal, etc.) |
| Next Hop IP Address | IP Address of the Next Hop (if applicable) |
| Next Hop Name | Name of the Next Hop (if applicable) |

**Step 4: Firewall**

Firewalls play a crucial role in overall network security. By default, Azure firewalls allow some essential ports for basic functionality. However, to open or close specific ports for your applications, you'll need to configure security rules or policies within the NSG you created earlier. Remember, a balance is key! While increased security is desirable, overly restrictive rules can hinder application functionality.

Firewall needs to be in sync with NSG rules for **ensuring consistent network access control**. This means that the firewall rules and NSG rules work together to define exactly which traffic is allowed to flow to and from your Azure resources. Any discrepancies between these rules could create security vulnerabilities or unintended access restrictions.

Documents to refer.

* https://learn.microsoft.com/en-us/azure/virtual-network/quick-create-portal
* <https://learn.microsoft.com/en-us/azure/virtual-network/virtual-network-manage-subnet?tabs=azure-portal>
* [Microsoft identity platform documentation - Microsoft identity platform | Microsoft Learn](https://learn.microsoft.com/en-us/entra/identity-platform/)
* <https://learn.microsoft.com/en-us/azure/virtual-network/concepts-and-best-practices>