

Microsoft Azure

# Implementing Virtual Networking

AZ-104 Microsoft Azure Administrator — Lab 04

Nicolas Portilla Gomez  
2-9-2026

## Table of Contents

executive summary .....	2
Key Outcomes .....	2
methodology.....	3
systems in scope .....	4
technical findings .....	6
Finding 1: Secure Network Traffic Control Was Implemented Using NSGs and ASGs.....	6
Finding 2: Virtual Networks and Subnets Were Deployed Using Infrastructure-as-Code .....	6
Finding 3: Network Segmentation Was Successfully Implemented .....	7
Finding 4: Private DNS Resolution Was Configured and Linked to Virtual Networks .....	7
Finding 5: Public DNS Resolution Was Configured and Validated.....	8
Finding 6: Separation of Public and Private DNS Was Maintained .....	8
screenshots & evidence .....	10
appendicies.....	16
References .....	17

## EXECUTIVE SUMMARY

This lab evaluated how virtual networking components are designed, deployed, and managed within Microsoft Azure to support secure and scalable cloud communication. The objective was to demonstrate how Azure Virtual Networks (VNets), subnets, network security groups (NSGs), and routing configurations enable controlled network segmentation, secure traffic flow, and connectivity between Azure resources.

During the lab, Azure virtual networking resources were created and configured to establish network boundaries and enforce traffic control policies. VNets and subnets were provisioned to segment resources logically, while network security groups were applied to regulate inbound and outbound traffic based on defined security rules. Additional configurations validated network connectivity and demonstrated how Azure networking constructs support isolation, security enforcement, and efficient resource communication. This approach illustrates how Azure virtual networking provides a foundational layer for secure cloud architecture and supports scalable, policy-driven network management.

## Key Outcomes

- Azure Virtual Networks enable logical isolation and structured network design for cloud resources.
- Subnetting improves network organization and supports layered security architectures.
- Network Security Groups enforce traffic control through rule-based filtering at the subnet and resource level.
- Proper virtual network configuration strengthens security posture and supports scalable, enterprise-grade cloud networking.

## METHODOLOGY

1. Review Azure virtual networking concepts, including virtual networks (VNets), subnets, and network security groups (NSGs), to understand their role in securing and organizing cloud network infrastructure.
2. Examine the structure and configuration of Azure virtual networks and subnets to understand how network segmentation and address space planning are implemented within Azure.
3. Deploy and configure Azure virtual networking resources to establish logical network boundaries and enable controlled communication between cloud resources.
4. Apply network security groups to subnets and resources to enforce inbound and outbound traffic rules based on defined security requirements.
5. Validate network configuration and connectivity through the Azure portal to confirm correct resource deployment, traffic flow behavior, and enforcement of network security rules.

## SYSTEMS IN SCOPE

System	Type	Purpose
Azure Subscription	Cloud Control Plane	Provides the administrative scope for deploying and managing Azure virtual networking resources and enforcing subscription-level policies.
Resource Group	Logical Container	Hosts virtual networking resources deployed during the lab and defines management and lifecycle boundaries.
Virtual Network (VNet)	Network Infrastructure	Provides logical network isolation and enables secure communication between Azure resources.
Subnet	Network Segmentation	Divides the virtual network into smaller address spaces to support organization, security, and traffic control.

System	Type	Purpose
<b>Network Security Group (NSG)</b>	Security Control	Enforces inbound and outbound traffic rules to regulate network access at the subnet or resource level.
<b>Azure Virtual Machines</b>	Compute Resource	Acts as network endpoints used to validate connectivity and security rule enforcement within the virtual network.

## **TECHNICAL FINDINGS**

### **Finding 1: Secure Network Traffic Control Was Implemented Using NSGs and ASGs**

#### **Description:**

Network traffic control was implemented using a combination of Network Security Groups (NSGs) and an Application Security Group (ASG). The ASG logically grouped web-facing resources, while the NSG enforced inbound and outbound traffic filtering at the subnet level. A custom inbound rule allowed only HTTP and HTTPS traffic from the ASG, and a custom outbound rule denied all internet-bound traffic. Default NSG rules were retained for internal virtual network communication.

#### **Evidence:**

Figure 1 - Figure 6

#### **Impact:**

This configuration enforces least-privilege network access, reduces attack surface, and strengthens network security through centralized, role-based traffic control.

### **Finding 2: Virtual Networks and Subnets Were Deployed Using Infrastructure-as-Code**

#### **Description:**

Virtual networks were deployed using Azure Resource Manager (ARM) templates to ensure consistent and repeatable infrastructure provisioning. The templates defined address spaces, subnet configurations, and locations declaratively. Existing virtual networks

were exported as ARM templates, parameterized, and reused to support standardized deployments across environments.

**Evidence:**

Figure 7 - Figure 9

**Impact:**

Infrastructure-as-code improves deployment consistency, minimizes configuration errors, and supports scalable cloud operations.

**Finding 3: Network Segmentation Was Successfully Implemented**

**Description:**

Virtual networks were configured with defined address spaces and multiple subnets to separate workloads such as shared services and databases. Subnet segmentation enabled logical isolation and supported layered security design within the network architecture.

**Evidence:**

Figure 10

**Impact:**

Subnet segmentation improves network organization, supports defense-in-depth strategies, and simplifies traffic management.

**Finding 4: Private DNS Resolution Was Configured and Linked to Virtual Networks**

**Description:**

A private DNS zone (*private.contoso2.com*) was created to support internal name resolution. Custom A records were added, and the zone

was linked to the Manufacturing virtual network. Auto-registration was disabled to maintain controlled DNS record management.

**Evidence:**

Figure 1, Figure 11

**Impact:**

Private DNS zones enable secure internal name resolution without exposing internal records to the public internet.

**Finding 5: Public DNS Resolution Was Configured and Validated**

**Description:**

A public DNS zone (*contoso2.com*) was configured to support external name resolution. DNS resolution was validated using the nslookup utility, confirming that public DNS records resolved to the expected IP address through Azure DNS.

**Evidence:**

Figure 2

**Impact:**

Public DNS configuration enables reliable external access to Azure-hosted services and confirms correct DNS publishing.

**Finding 6: Separation of Public and Private DNS Was Maintained**

**Description:**

The environment maintained a clear separation between private DNS zones for internal resolution and public DNS zones for external access.

This design prevents internal DNS records from being exposed publicly while still supporting external name resolution.

**Evidence:**

Figure 1, Figure 2, Figure 11

**Impact:**

Separating DNS scopes supports security best practices and reduces the risk of unintended information disclosure.

# SCREENSHOTS & EVIDENCE

This screenshot shows the Microsoft Azure portal interface for managing a private DNS zone. The URL in the address bar is `private.contoso2.com`. On the left, the navigation menu is open, showing options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings, DNS Management, Recordsets, and Virtual Network Links. The 'Virtual Network Links' option is selected. The main content area displays a table of virtual network links. There is one entry: 'manufacturing-link' with 'Link Status' as 'Completed', 'Virtual Network' as 'ManufacturingVnet', 'Auto-Registration' as 'Disabled', and 'Fallback to internet' as 'Disabled'. A search bar at the top right says 'Search resources, services, and docs (G+J)'. The top right corner shows the user's name 'nicolasportilla2011@ho...' and other account details.

Figure 1: Private DNS zone `private.contoso2.com` linked to ManufacturingVnet with link status Completed and Auto-registration Disabled, enabling controlled private DNS resolution within the VNet.

This screenshot shows the Microsoft Azure portal interface for managing a DNS recordset. The URL in the address bar is `contoso2.com_1770685171210|Overview > contoso2.com`. The navigation menu is open, showing options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings, DNS Management, Recordsets, and DNSSC. The 'Recordsets' option is selected. The main content area shows a table of record sets. There is one entry: 'www' with 'Name' as 'www.contoso2.com', 'Type' as 'A', 'TTL' as '3600', and 'IP Address' as '10.1.1.4'. Below the table, it says 'Minimum TTL: 300' and 'Serial number: 1'. A PowerShell window is overlaid on the page, showing the command `nslookup www.contoso2.com ns1-02.azure-dns.com.` The output of the command is displayed, showing the IP address 10.1.1.4. The top right corner shows the user's name 'nicolasportilla2011@ho...' and other account details.

Figure 2: `nslookup` test confirming `www.contoso2.com` resolves successfully (via Azure DNS) to the expected IP address, validating DNS record configuration and name resolution.

The screenshot shows the Azure portal interface for managing an Application Security Group (ASG). The top navigation bar includes links for resources, services, and docs (G+), Copilot, and user information (nicolasportilla2011@...). The main title is "List security rules for this Application Security Group." with a count of "+2".

**Overview** (selected)

- Activity log
- Access control (IAM)
- Tags
- Resource visualizer

**Essentials**

Resource group ( <a href="#">move</a> )	Virtual Network
<a href="#">az104-rg4</a>	
Location ( <a href="#">move</a> )	
South Central US	
Subscription ( <a href="#">move</a> )	
<a href="#">Azure subscription 1</a>	
Subscription ID	3fa04a5e-627e-473a-99fa-fa113375f406
Provisioning state	Succeeded

Tags ([edit](#))  
[Add tags](#)

Showing the network interfaces linked to this application security group. Only the primary IP address of each network interface is shown. You can add or remove one or more network interfaces associated with asg-web in this virtual network.

+ Add | X Remove

<input type="checkbox"/>	Private IP address ↑	Network int... ↑	Attached to ↑	Resource type ↑
--------------------------	----------------------	------------------	---------------	-----------------

Add or remove favorites by pressing **Ctrl+Shift+F**

Figure 3: Application Security Group asg-web created successfully in resource group az104-rg4, providing an application-based grouping mechanism for NSG rules.

**Overview**

Resource group (move) : az104-rg4  
Location : South Central US  
Subscription (move) : Azure subscription 1  
Subscription ID : 3fa04a5e-627e-473a-99fa-fa113375f406  
Tags (edit) : Add tags

**Inbound Security Rules**

Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
100	AllowASG	80,443	TCP	ASG-WEB	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerIn...	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

**Outbound Security Rules**

Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
4096	DenyInternetOutbound	Any	Any	Any	Internet	Deny
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow

Figure 4: Network Security Group myNSGSecure showing custom rules (1 inbound, 1 outbound) and association with 1 subnet, confirming NSG deployment and scope.

**Inbound security rules**

Network security group security rules are evaluated by priority using the combination of source, source port, destination port and direction as an existing rule. You can't delete default security rules, but you can override them with rules

**AllowASG**

Source : Application security group  
Source application security groups : ASG-WEB  
Source port ranges : \*  
Destination : Any  
Service : Custom  
Destination port ranges : 80,443  
Protocol : TCP

Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓
100	AllowASG	80,443	TCP
65000	AllowVnetInBound	Any	Any
65001	AllowAzureLoadBalancerIn...	Any	Any
65500	DenyAllInBound	Any	Any

Figure 5: Inbound NSG rule AllowASG configured to allow TCP ports 80 and 443 using ASG-WEB as the source, restricting web access to approved application resources.

The screenshot shows the Microsoft Azure portal interface for managing Network Security Groups (NSGs). The left sidebar navigation shows the NSG 'myNSGSecure'. The main content area displays the 'Outbound security rules' section. A modal window titled 'DenyInternetOutbound' is open, detailing the configuration for this rule. The rule is set to deny traffic to the 'Internet' service tag from 'Any' source port range (all ports) to 'Any' destination port range (all ports), using 'Any' protocol. The 'Protocol' dropdown is set to 'Any', and the 'Source' dropdown is also set to 'Any'. The 'Save' button is visible at the bottom of the modal.

Figure 6: Outbound NSG rule DenyInternetOutbound configured to deny traffic to the Internet service tag, limiting outbound connectivity and enforcing controlled egress.

The screenshot shows the Microsoft Azure portal interface for managing Resource Groups. The left sidebar navigation shows the Resource Group 'az104-rg4'. The main content area displays the 'Essentials' section under the 'Resources' tab. It lists two resources: 'CoreServicesVnet' and 'ManufacturingVnet', both of which are 'Virtual network' type resources located in 'South Central US' and 'West Europe' respectively. The 'JSON View' link is visible at the top right of the table.

Figure 7: ARM template file showing declarative configuration for ManufacturingVnet, including address space definitions and subnet resources.

```

1  {
2     "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
3     "contentVersion": "1.0.0.0",
4     "parameters": {
5         "virtualNetworks_ManufacturingVnet_name": {
6             "defaultValue": "ManufacturingVnet",
7             "type": "String"
8         }
9     },
10    "variables": {},
11    "resources": [
12        {
13            "type": "Microsoft.Network/virtualNetworks",
14            "apiVersion": "2023-05-01",
15            "name": "[parameters('virtualNetworks_ManufacturingVnet_name')]",
16            "location": "westeurope",
17            "properties": {
18                "addressSpace": {
19                    "addressPrefixes": [
20                        "10.30.0.0/16"
21                    ]
22                },
23                "encryption": {
24                    "enabled": false,
25                    "enforcement": "AllowUnencrypted"
26                },
27                "subnets": [
28                    {
29                        "name": "SensorSubnet1",
30                        "id": "[resourceId('Microsoft.Network/virtualNetworks/subnets', parameters('virtualNetworks_ManufacturingVnet_name'), 'sensorsubnet1'))",
31                        "properties": {
32                            "addressPrefixes": [
33                                "10.30.10.0/24"
34                            ],
35                            "delegations": [],
36                            "privateEndpointNetworkPolicies": "Disabled",
37                            "privateLinkServiceNetworkPolicies": "Enabled"
38                        }
39                    }
40                ]
41            }
42        }
43    ]
44}

```

Figure 8: Downloaded ARM template edited locally to support customization and parameterized deployment, demonstrating infrastructure-as-code reuse.

The screenshot shows the Azure portal interface for a 'CoreServicesVnet' virtual network. On the left, there's a navigation sidebar with options like Home, CoreServicesVnet, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings, Monitoring, Automation, Tasks, and Export template. The 'Export template' option is highlighted. The main area shows the 'CoreServicesVnet | Export template | Template' view. It includes tabs for ARM template, Bicep, and Terraform. Below the tabs are buttons for Download, Copy, and Deploy. A tooltip explains how to export resources by selecting them in the Overview view and using the 'Export template' option. There's also a checked checkbox for 'Include parameters'. The central part of the screen displays the ARM template code, which is identical to the one shown in Figure 8. The right side of the screen shows a preview of the virtual network resources.

Figure 9: Azure portal Export template view for CoreServicesVnet with parameters included, supporting standardized redeployment.

Home > Network foundation | Virtual networks

## Create virtual network ...

Validation passed

Basics Security IP addresses Tags **Review + create**

Subscription	Azure subscription 1
Resource Group	az104-rg4
Name	CoreServicesVnet
Region	South Central US
<b>Security</b>	
Azure Bastion	Disabled
Azure Firewall	Disabled
Azure DDoS Network Protection	Disabled
<b>IP addresses</b>	
Address space	10.20.0.0/16 (65,536 addresses)
Subnet	SharedServicesSubnet (10.20.10.0/24) (256 addresses)
Subnet	DatabaseSubnet (10.20.20.0/24) (256 addresses)

Previous Next Create Download a template for automation Give feedback

Figure 10: Virtual network creation summary displaying address space 10.20.0.0/16 and two subnets (SharedServicesSubnet and DatabaseSubnet), confirming subnet segmentation.

Microsoft Azure

private.contoso2.com | Recordsets

Search resources, services, and docs (G+)

nicolasportilla2011@ho... DEFAULT DIRECTORY (NICOLAS...)

Home > private.contoso2.com

+ Add ⌂ Refresh ⌂ Delete ⌂ Give feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Resource visualizer Settings DNS Management Recordsets Virtual Network Links Monitoring Automation Help

A record set is a collection of records in a zone that have the same name and are the same type. Record Sets will be automatically fetched in batches of 100 as you scroll through the existing record sets. [Learn more](#)

Search Fetched 2 record set(s). 0 record sets selected

Name	Type	TTL	Value	Auto registered
@	SOA	3600	Email: azureprivatedns-host.microsoft.com Host: azureprivatedns.net Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 10 Serial number: 1	False
sensorvm	A	3600	10.1.1.4	False

Add or remove favorites by pressing **Ctrl+Shift+F**

Figure 11: Private DNS zone private.contoso2.com records showing default SOA record and custom A record (sensorvm) mapped to private IP 10.1.1.4, confirming internal DNS resolution.

## **APPENDICES**

**REFER TO GITHUB FOR TEMPLATE.JSON AND PARAMETERS.JSON**

## **REFERENCES**

- [1] Microsoft, "Microsoft Learning," 2025. [Online]. Available:  
[https://microsoftlearning.github.io/AZ-104-MicrosoftAzureAdministrator/Instructions/Labs/LAB\\_04-Implement\\_Virtual\\_Networking.html](https://microsoftlearning.github.io/AZ-104-MicrosoftAzureAdministrator/Instructions/Labs/LAB_04-Implement_Virtual_Networking.html). [Accessed 9 February 2026].