Hub Spoke Network

A Hub Spoke is a network design where a Central Virtual Network (VNet) acts as a Hub which acts as a point of connectivity for several other VNets which are the Spokes which are the actual VMs and Application where all the work is done. Spokes cannot talk to each other directly and must do everything through the Hub. This network design is useful because the Hub hosts all shares services such as DNS, Firewall, VPN, Bastion etc. making it possible for Spokes to share the services and thus reducing costs.

Common Terminologies

IP Address: A numerical label given to every device in a network that uses Internet Protocol.
Example: 127.105.14.23.
The first part represents the Network, **127 is loopback block meaning it refers to the device itself**. The second part further refines the network called subnet. Depending on the Subnet Masking the third part can be either part of a network or a part of the Host ID. The fourth part is the Host ID which identifies the specific device in the network. More details.

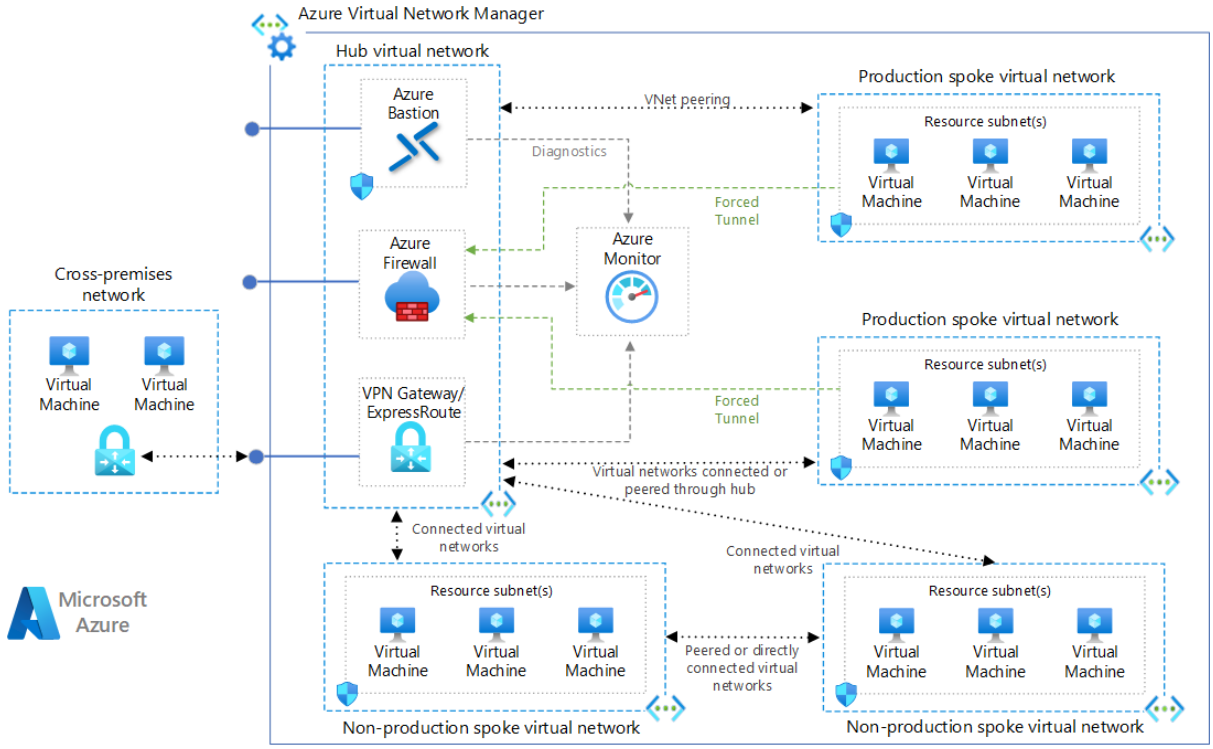
DNS: DNS is the naming system that converts human-readable into IP Addresses. Like saving a Phone number with an easier name to remember. Example for google.com it's IP Address is 142.250.191.174.
However, it should also be noted that most companies use Virtual Hosting so a single IP Address can host many different websites and services. Also google shifts traffic between many IP Addresses due to the large request. Also, many networks use dynamic IP Addresses meaning that the IP Addresses change. More Details.

Subnet Masking: A subdivision of an IP network which allows us to organize resources. /16 means the first 2 parts of an IP is for the Network. /24 means the 3 parts of an IP is for the Network. More Details.

Firewall: A security device that monitors and filters incoming and outgoing network traffic based on security rules. More Details.

VPN Gateway: A specific type of virtual network gateway that sends encrypted traffic between an Azure virtual network and an on-premises location over the public Internet. More Details.

Bastion: Azure Bastion is a service that is used to securely connect to virtual machines via private IP address. More Details.



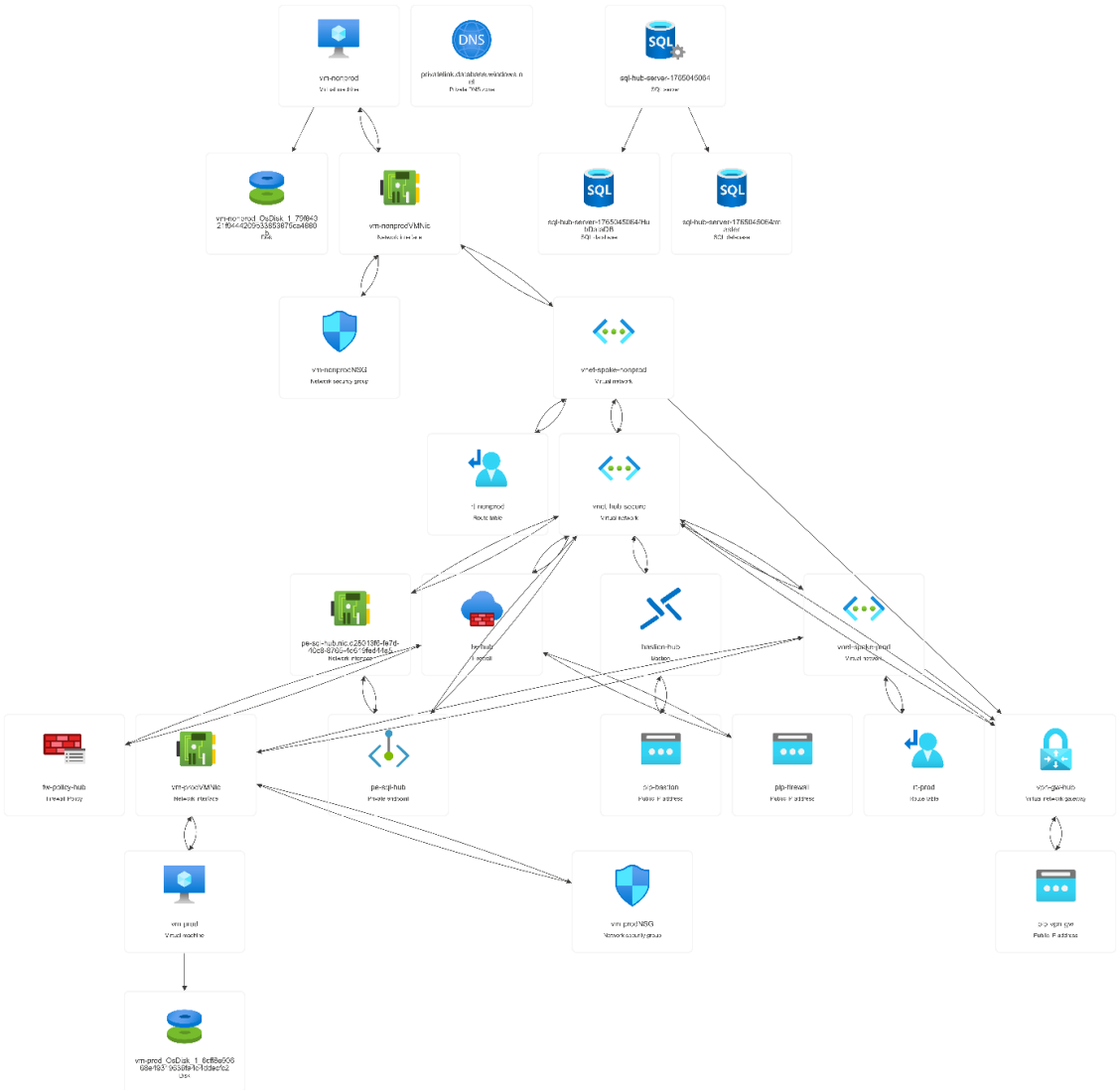
Microsoft Azure Hub-Spoke Architecture

Project Overview

This project implements a secure, scalable Hub-and-Spoke network topology in the Azure Japan East region using Bash automation scripts. The design utilizes a central "Hub" Virtual Network (VNet) to host shared services, while "Spoke" VNets host isolated workloads, reducing costs and administrative overhead.

Key Highlights:

- Implemented centralized security with Azure Firewall (AZFW_VNet SKU) enforcing User-Defined Routes (UDRs) to inspect traffic between spokes.
- Deployed a Azure SQL Database with Private Link and Private DNS zones, eliminating public internet exposure across all VNets
- Established hybrid connectivity with VPN Gateway (VpnGw1) and secure administrative access via Azure Bastion.
- Validated cross-spoke connectivity with no packet loss through firewall (10.0.1.4)



Project Topology

Hub

The **Hub VNet** (vnet-hub-secure) is the center of the network hosting all of the following shared services and apps:

- **Azure Firewall** (fw-hub) is what controls the traffic between the different VNets. It has a public IP for going to the internet and a private IP for the internal traffics.
- **VPN Gateway** (vpn-gw-hub) creates an encrypted path over the internet for the VNets inside the network keeping them anonymous.
- **Bastion** (bastion-hub) assures secure connection to virtual machines via private IP addresses.
- **SQL Private Endpoint** (pe-sql-hub) is the virtual network interface for the SQL Database in the network. Even though it is in the Azure Cloud, this endpoint forces it to appear physically in the Hub VNet. In this case the IP is 10.0.4.x.

Spokes

There are two workload vnets or **Spokes** in this network. One is the **Non-Production** (vnet-spoke-nonprod) which is the development server, and the **Production** (vnet-spoke-prod) which is the production server which releases the finalized product. They cannot talk to each other and have totally separate networks. Incase there were multiple Production and Non-Production VNets they had to talk to each other through the hub.

The VNet Peering connects the hub to the spokes. The Gateway Transit allows the spokes to use the hub's VPN Gateway to use the Internet.
All of this traffic data is forced through the firewall making it tough to hack. Every traffic has to go through 10.0.1.4 which is the firewall IP.
I have also created a Private DNS Zones for the SQL Server which links all the VNets. When any VM tries to reach the SQL server by name, it ensures the Private IP is resolved, keeping traffic completely off the public internet.

Scripts

Hub.sh: Creates the Network Hub-Spoke-Tokyo in Tokyo Region (Japan East). It then divides the Hub VNet (10.0.0.0/16) into subnets

- Firewall Subnet: 10.0.1.0/24
- VPN Gateway Subnet: 10.0.2.0/24
- Bastion Subnet: 10.0.3.0/24
- Database Subnet: 10.0.4.0/24

All these also have public IPs enabling them to communicate with the outside world.

It also creates the Production and Non-Production VMs with the IPs.

- Production: 10.1.0.0/16
- Non Production: 10.2.0.0/16

Finally, it establishes connection between them.

It has different failsafe commands such as

- The script crashes immediately if any command fails or if a variable is missing, preventing half-broken deployments.
- Check all Prerequisites

Routing.sh: This script enforces every traffic through the firewall created in the hub.sh. The VMs must go through the firewall to go anywhere. It binds these rules into the Spokes Subnets.

Similarly, it has failsafe checks which are making sure there is a firewall and it has a private IP.

Database.sh: This script deploys the database which is the application of this network. The server is completely hidden from public internet. It checks for an Azure Key Vault to retrieve the password. It creates a Private Endpoint in the Hub VNet enabling it to have a local LAN IP. There is a Private DNS Zone which links it to the Hub and Spokes. This ensures that the private IP is given and not the Public IP for Azure Cloud. There is also a function that detects the local PC's IP and adds a temporary bypass through the firewall to the SQL Server, enabling us to access the sql server.

Deployment Verification Results:

```
═══════════════════════════════════════════════════
   AZURE HUB-AND-SPOKE ARCHITECTURE - DEPLOYMENT SUMMARY
═══════════════════════════════════════════════════

  Region: Japan East (Tokyo)
    Architecture: Hub-and-Spoke with Centralized Security

--- INFRASTRUCTURE COMPONENTS ---
✅ Hub VNet: 10.0.0.0/16
✅ Prod Spoke: 10.1.0.0/16
✅ NonProd Spoke: 10.2.0.0/16
✅ Azure Firewall: 10.0.1.4
✅ VPN Gateway: Succeeded
✅ Azure Bastion: Succeeded
✅ SQL Database: Private Link Enabled

--- CONNECTIVITY TEST ---
2 received

--- SECURITY POSTURE ---
  Zero Trust: VMs have no public IPs
  Centralized Security: All traffic via Azure Firewall
  Private Database: SQL accessible only via Private Link
  Secure Access: Azure Bastion for admin access

═══════════════════════════════════════════════════
```

Deployment Summary

```
Asus@TAUSIF-LAPTOP MINGW64 /e/Projects/Hub_Spoke_Network
$ echo ""; echo "=== Hub Peerings ==="; az network vnet peering list -g Hub-Spoke-Tokyo --vnet-name vnet-hub-secure --query "[].{Name:name,PeeringState:peeringS
tate,UseRemoteGateways:useRemoteGateways,AllowGatewayTransit:allowGatewayTransit}" -o table; echo ""; echo "=== Prod Spoke Peerings ==="; az network vnet peerin
g list -g Hub-Spoke-Tokyo --vnet-name vnet-spoke-prod --query "[].{Name:name,PeeringState:peeringState,UseRemoteGateways:useRemoteGateways,AllowGatewayTransit:a
llowGatewayTransit}" -o table; echo ""; echo "=== NonProd Spoke Peerings ==="; az network vnet peering list -g Hub-Spoke-Tokyo --vnet-name vnet-spoke-nonprod --
query "[].{Name:name,PeeringState:peeringState,UseRemoteGateways:useRemoteGateways,AllowGatewayTransit:allowGatewayTransit}" -o table 2>/dev/null

=== Hub Peerings ===
Name            PeeringState    UseRemoteGateways    AllowGatewayTransit
--------------  --------------  -------------------  ---------------------
Hub-to-Prod     Connected       False                True
Hub-to-NonProd  Connected       False                True

=== Prod Spoke Peerings ===
Name         PeeringState    UseRemoteGateways    AllowGatewayTransit
-----------  --------------  -------------------  ---------------------
Prod-to-Hub  Connected       True                 False

=== NonProd Spoke Peerings ===
Name            PeeringState    UseRemoteGateways    AllowGatewayTransit
--------------  --------------  -------------------  ---------------------
NonProd-to-Hub  Connected       True                 False
```

VNet Peering Status

```
=== Routing Configuration Summary ===

Firewall Private IP:
Name       PrivateIP
---------  -----------
fw-config  10.0.1.4

Route Tables:
Name
----------
rt-nonprod
rt-prod

Prod Route Details:
AddressPrefix    NextHopType       NextHopIP
---------------  ----------------  -----------
0.0.0.0/0        VirtualAppliance  10.0.1.4

NonProd Route Details:
AddressPrefix    NextHopType       NextHopIP
---------------  ----------------  -----------
0.0.0.0/0        VirtualAppliance  10.0.1.4

Subnet Associations:
{
  "RouteTable": "/subscriptions/02845afb-e239-407b-9ddf-a1a43529bbb2/resourceGroups/Hub-Spoke-Tokyo/providers/Microsoft.Network/routeTables/rt-prod",
  "Subnet": "vnet-spoke-prod/default"
}
{
  "RouteTable": "/subscriptions/02845afb-e239-407b-9ddf-a1a43529bbb2/resourceGroups/Hub-Spoke-Tokyo/providers/Microsoft.Network/routeTables/rt-nonprod",
  "Subnet": "vnet-spoke-nonprod/default"
}
```

Ln 86, Col 3 (1214 selected) Spaces: 2 UTF-8 LF { } Shell Script Go Live Prettier

Routing Configuration (UDR)

```
Asus@TAUSIF-LAPTOP MINGW64 /e/Projects/Hub_Spoke_Network
$ az sql server show -g Hub-Spoke-Tokyo -n sql-hub-server-1765045064 --query "{Name:name,PublicNetworkAccess:publicNetworkAccess,State:state}" -o table
Name                       PublicNetworkAccess    State
-------------------------  ---------------------  -------
sql-hub-server-1765045064  Disabled               Ready

Asus@TAUSIF-LAPTOP MINGW64 /e/Projects/Hub_Spoke_Network
$ az network private-endpoint show -g Hub-Spoke-Tokyo -n pe-sql-hub --query "{Name:name,ProvisioningState:provisioningState,PrivateIP:customDnsConfigs[0].ipAddr
esses[0]}" -o table
Name        ProvisioningState
----------  -------------------
pe-sql-hub  Succeeded

Asus@TAUSIF-LAPTOP MINGW64 /e/Projects/Hub_Spoke_Network
$ az network private-dns link vnet list -g Hub-Spoke-Tokyo --zone-name privatelink.database.windows.net --query "[].{Name:name,RegistrationEnabled:registrationE
nabled,ProvisioningState:provisioningState}" -o table
Name          RegistrationEnabled    ProvisioningState
------------  ---------------------  -------------------
link-hub      False                  Succeeded
link-nonprod  False                  Succeeded
link-prod     False                  Succeeded
```

Database Security & Private Link