# Deployment Architecture

The Inspire Brands deployment on Azure will utilize a 3-Tier architecture with an Application GW as the load balancer. Tier 1 will utilize API Management deployed internally on the APIM Subnet. Tier 2 will utilize the Azure Kubernetes Service with the containers utilizing Azure CNI Networking. The data tier will utilize private endpoints for Azure SQL, Cosmos DB, Key Vault, Azure Container Registry, and Service Bus. The data tier will also utilize the Premium Tier of Azure Cache injected on the subnet.

The following are the entry points to the Azure Architecture

* External Users access the Azure APIM service through the Application Gateway’s public IP.
* On-Premises Users access the environment through the VPN Gateway. This will allow access to the Jump Box or the Application Gateway through the internal IP.
* Outside of the Corporate Subnet you can access the Jump Box through the Azure Bastion Service
* There will also be a data pipeline that accesses the POC systems in the store and inserts the data into Cosmos DB

A close up of a map

Description automatically generated

# Virtual Network Design

The virtual network will be split into 6 subnets as described below.

## Application Gateway Subnet

The Application Gateway must sit on a dedicated subnet with at least a /28 address space. Application Gateway can scale up and down the number of instances according to the load on the gateway so your address space must account for the scale.

<https://docs.microsoft.com/en-us/azure/application-gateway/application-gateway-autoscaling-zone-redundant>

<https://docs.microsoft.com/en-us/azure/application-gateway/configuration-overview#size-of-the-subnet>

|  |  |  |
| --- | --- | --- |
| Subnet Name | Address Space | NSG Name |
| Application Gateway SN | x.x.x.x/26 | <No NSG Allowed on Subnet> |

## APIM Subnet

The APIM Subnet will contain the Azure API Management service in Internal Mode:

<https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-integrate-internal-vnet-appgateway>

|  |  |  |
| --- | --- | --- |
| Subnet Name | Address Space | NSG Name |
| APIM-SN | x.x.x.x/24 | APIMNSG |

## AKS Subnet

The AKS Subnet will contain the Azure Kubernetes service nodes utilizing Azure CNI Networking:

<https://docs.microsoft.com/en-us/azure/aks/configure-azure-cni>

|  |  |  |
| --- | --- | --- |
| Subnet Name | Address Space | NSG Name |
| AKS-SN | x.x.x.x/16 | AKSNSG |

## Backend Subnet

The Backend Subnet will hos the data tier of the application. For the PaaS offerings that support Private Endpoints we will place those private IPs on this subnet. The Azure Cache offering doesn’t currently support Private Endpoints but the premium tiers support VNet injection.

<https://docs.microsoft.com/en-us/azure/private-link/private-link-overview>

<https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-how-to-premium-vnet>

|  |  |  |
| --- | --- | --- |
| Subnet Name | Address Space | NSG Name |
| Backend-SN | x.x.x.x/16 | AKSNSG |

## Azure Bastion Subnet

The Azure Bastion solution must sit on a dedicated subnet with at least a /28 subnet address. This service allows you to access the VMs in a virtual network securely without a VPN or public IP address associated with the VMs.

|  |  |  |
| --- | --- | --- |
| Subnet Name | Address Space | NSG Name |
| AzureBastion | x.x.x.x/28 | <No NSG Allowed on Subnet> |

## Management Subnet

The Management Subnet will host one or more Jump Boxes for administration of the Azure Environment.

|  |  |  |
| --- | --- | --- |
| Subnet Name | Address Space | NSG Name |
| Management-SN | x.x.x.x/26 | ManagementNSG |

# Network Security Group Design

The traffic allowed between the subnets is controlled via Network Security Groups. On each NSG you can define both inbound and outbound rules that define what TCP/UDP traffic is allowed.

<https://docs.microsoft.com/en-us/azure/virtual-network/security-overview>

APIM NSG:

| **Source / Dest Port(s)** | **Direction** | **Protocol** | [**Service Tags**](https://docs.microsoft.com/en-us/azure/virtual-network/security-overview#service-tags) **Source / Destination** | **Purpose (\*)** |
| --- | --- | --- | --- | --- |
| \* /443 | Inbound | TCP | AppGW Subnet | Client communication to API Management |
| \* / 3443 | Inbound | TCP | ApiManagement / VIRTUAL\_NETWORK | Management endpoint for Azure portal and PowerShell |
| \* / 443 | Outbound | TCP | VIRTUAL\_NETWORK / Storage | Dependency on Azure Storage |
| \* / 443 | Outbound | TCP | VIRTUAL\_NETWORK / AzureActiveDirectory | [Azure Active Directory](https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-aad) (where applicable) |
| \* / 5671, 5672, 443 | Outbound | TCP | VIRTUAL\_NETWORK / EventHub | Dependency for [Log to Event Hub policy](https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-log-event-hubs) and monitoring agent |
| \* / 445 | Outbound | TCP | VIRTUAL\_NETWORK / Storage | Dependency on Azure File Share for [GIT](https://docs.microsoft.com/en-us/azure/api-management/api-management-configuration-repository-git) |
| \* / 1886 | Outbound | TCP | VIRTUAL\_NETWORK / AzureCloud | Needed to publish Health status to Resource Health |
| \* / 443 | Outbound | TCP | VIRTUAL\_NETWORK / AzureMonitor | Publish [Diagnostics Logs and Metrics](https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-use-azure-monitor) |
| \* / 25 | Outbound | TCP | VIRTUAL\_NETWORK / INTERNET | Connect to SMTP Relay for sending e-mails |
| \* / 587 | Outbound | TCP | VIRTUAL\_NETWORK / INTERNET | Connect to SMTP Relay for sending e-mails |
| \* / 25028 | Outbound | TCP | VIRTUAL\_NETWORK / INTERNET | Connect to SMTP Relay for sending e-mails |
| \* / 6381 - 6383 | Inbound & Outbound | TCP | VIRTUAL\_NETWORK / VIRTUAL\_NETWORK | Access Redis Service for [Rate Limit](https://docs.microsoft.com/en-us/azure/api-management/api-management-access-restriction-policies#LimitCallRateByKey) policies between machines |
| \* / \* | Inbound & Outbound | TCP | \* | Deny All |

**AKS NSG**

A network security group filters traffic for VMs, such as the AKS nodes. As you create Services, such as a LoadBalancer, the Azure platform automatically configures any network security group rules that are needed. Don't manually configure network security group rules to filter traffic for pods in an AKS cluster. Define any required ports and forwarding as part of your Kubernetes Service manifests, and let the Azure platform create or update the appropriate rules. You can also use network policies, as discussed in the next section, to automatically apply traffic filter rules to pods.

**Backend NSG**