Azure Stack to Azure Hybrid Connection with WebApp

Technical guidance

Contents

[1 Overview 2](#_Toc522864046)

[1.1 Context and Considerations 2](#_Toc522864047)

[1.2 When to Use This Pattern 2](#_Toc522864048)

[2 Reference Architecture 3](#_Toc522864049)

[2.1 Azure Resources 3](#_Toc522864050)

[2.2 Azure Stack Resources 4](#_Toc522864051)

[3 Prerequisites 5](#_Toc522864052)

[3.1 Before you begin 5](#_Toc522864053)

[4 Deploying Azure Stack Resources 6](#_Toc522864054)

[4.1 Preparing Parameters (Azure Stack) 6](#_Toc522864055)

[4.2 Deploying Template using Azure PowerShell (Azure Stack) 9](#_Toc522864056)

[4.3 Deploying Template using Azure CLI Commands (Azure Stack) 11](#_Toc522864057)

[5 Deploying Azure Resources 16](#_Toc522864058)

[5.1 Preparing Parameters (Azure) 16](#_Toc522864059)

[5.2 Deploying Template using Azure PowerShell (Azure) 17](#_Toc522864060)

[5.3 Deploying Template using Azure CLI (Azure) 19](#_Toc522864061)

[5.4 Configuring Azure Connection 22](#_Toc522864062)

[6 Verifying Deployment 25](#_Toc522864063)

[6.1 Azure Connection 25](#_Toc522864064)

[6.2 WebApp Application Settings 26](#_Toc522864065)

[6.3 WebApp Application files 26](#_Toc522864066)

[6.4 WebApp Connectivity 29](#_Toc522864067)

[7 Appendix 30](#_Toc522864068)

[7.1 Configuring BGP for Azure Stack Development Kit Only 30](#_Toc522864069)

[7.2 Using the North Wind WebApp 34](#_Toc522864070)

[7.3 Verify Data in Database 36](#_Toc522864071)

1. Overview

This reference article details environmental requirements and steps for setting up Azure to Azure Stack Hybrid Connections.

Microsoft is the only cloud provider that offers a truly consistent hybrid cloud platform, including a consistent hybrid networking experience. Setting a hybrid connection between an Azure Virtual Network and Azure Stack Virtual Network is simple, using the same process in each cloud.

The hybrid network provides secure access between a virtual network in Azure and a virtual network in Azure Stack. Endpoints in Azure, including App Service applications linked to a virtual network, can communicate with endpoints in Azure Stack as if they were on the same network.

* 1. Context and Considerations

There are some distinctions between one-node Azure Stack Development Kit (ASDK), and multi-node Azure Stack Integrated System (ASIS).

ASDK utilizes a public IP address, while maintaining its own VM with a separate and defined private network.

Azure Stack Integrated System integrates with your datacenter and has an entire IP address range to delegate to the system during installation. This requires some specialized setup and configuration for the Azure stack isolated environment. Software-defined networking requires only four cables connecting an Azure Stack machine to the outside network.

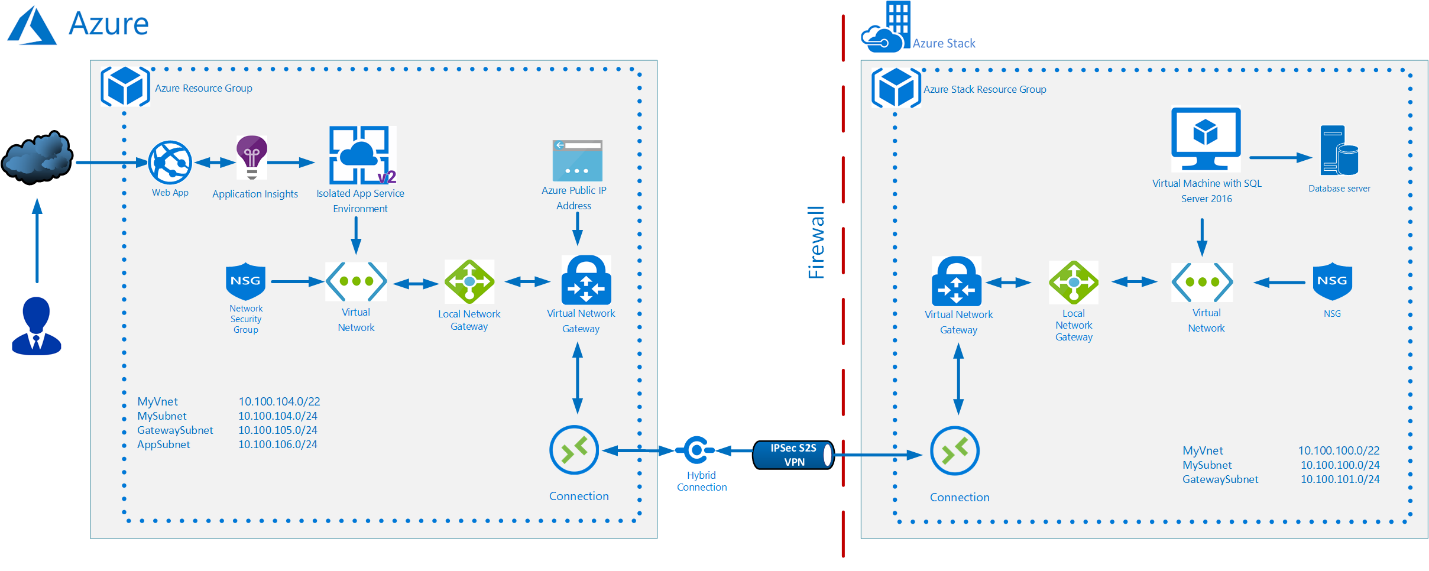
* 1. When to Use This Pattern

**Use hybrid cloud resources to implement a Hybrid Connection**

* Hybrid Connectivity is a foundational pattern that allows you securely access resources in an Azure Stack deployment from Azure.
* Certain data must live on-premises because of privacy or regulatory requirements.
* Maintain a legacy system while utilizing cloud-scaled app deployment.

1. Reference Architecture

This section details the reference architecture that can be used as a guidance to implement the offer.



This architecture consists of the following components:

* 1. Azure Resources
* **Azure App Services**. Build, deploy, and scale enterprise-grade web, mobile, and serverless compute applications and as well as leveraging RESTful APIs running on any platform with Platform-as-a-service (PaaS) offerings. For more information about Azure App Services see [Microsoft Azure App Services Overview](https://azure.microsoft.com/en-us/services/app-service/).
* **Azure Virtual Network.** [Azure Virtual Network](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview) enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks.
  + **Application Subnet.** Dividing the Azure Virtual Network into two or more logical, IP subdivisions via subnets provides a custom private IP address space using public and private (RFC 1918) addresses. This subnet will be where many of the resources will be deployed.
  + **Gateway Subnet.** The gateway subnet is part of the virtual network IP address range specified when configuring the virtual network, and contains the IP addresses that the [virtual network gateway](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-create-site-to-site-rm-powershell) resources and services use.
* **Azure Local Network Gateway.** The local network gateway typically refers to the on-premises location. Azure refers to the site name and specifies the IP address of the local VPN device to connect to.
* **Azure Virtual Network Gateway.** The Azure Virtual Network Gateway acts as a [Site-to-Site VPN gateway](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-gateway-settings) connection that is used to connect the on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel.
* **Azure Public IP.** [Public IP addresses](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-ip-addresses-overview-arm#public-ip-addresses) allow Internet resources to communicate inbound to Azure resource, and enable Azure resources to communicate outbound to Internet and public-facing Azure services with an IP address assigned to the resource.
* **Azure Point to Site Application VPN.** A [Point-to-Site (P2S)](https://docs.microsoft.com/en-us/azure/vpn-gateway/point-to-site-about) VPN connection allows a secure connection to the virtual network from an individual client computer. This solution is useful for telecommuters needing to connect to Azure VNets from a remote location.
  1. Azure Stack Resources
* **Azure Stack IaaS for Hosting a Microsoft SQL Server VM.** Use the same application model, self-service portal, and APIs enabled by Azure. [Azure Stack IaaS](https://azure.microsoft.com/en-us/overview/azure-stack/benefits/) allows for a broad range of open source technologies for consistent hybrid cloud deployments.
* **Azure Stack Virtual Network.** The Azure Stack Virtual Network, works exactly like the [Azure Virtual Network](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview), and enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks.
  + **Application Subnet.** Dividing the Azure Virtual Network into two or more logical, IP subdivisions via subnets provides a custom private IP address space using public and private (RFC 1918) addresses. This subnet will be where the majority of the resources will be deployed.
  + **Gateway Subnet.** The gateway subnet is part of the virtual network IP address range specified when configuring the virtual network, and contains the IP addresses that the [virtual network gateway](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-create-site-to-site-rm-powershell) resources and services use.
* **Azure Stack Virtual Network Gateway.** Send network traffic between Azure virtual network and an on-premises site by creating a [virtual network gateway.](https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-vpn-gateway-about-vpn-gateways)
* **Azure Stack Local Network Gateway.** [The local network gateway typically refers to the on-premises location. Azure refers to the site name and specifies the IP address of the local VPN device to connect to.](https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-network)

**Azure Stack Public IP.** The Azure Stack [Public IP addresses](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-ip-addresses-overview-arm#public-ip-addresses) work like the Azure Public IP addresses, allowing Internet resources to communicate inbound to Azure resource, and enable Azure resources to communicate outbound to Internet and public-facing Azure services with an IP address assigned to the resource. As note, please work with the Hardware OEM Partners to make Azure Stack services (such as the portals, Azure Resource Manager, DNS, etc.) available to external networks.

1. Prerequisites

**Azure Stack**

* Firewall and or router appliance needs to know how to route traffic to and from Azure Stack environment
* An Azure Stack Environment.

For information on how to deploy Azure Stack Development Kit see [ASDK-Install](https://docs.microsoft.com/azure/azure-stack/asdk/asdk-install)

* Azure Stack environment has SQLRP deployed and configured.

For information on how to deploy Azure Stack Development Kit see [Azure Stack-SQL-Resource-Provider-Deploy](https://docs.microsoft.com/azure/azure-stack/azure-stack-sql-resource-provider-deploy)

* SQL Server 2016 image added to your Azure Stack Marketplace.

For information on how to add Marketplace images from Azure Marketplace see [Adding-Images](https://docs.microsoft.com/azure/azure-stack/asdk/asdk-register)

* Plans, Offers and Quotas Configured.

For information on how to configure Quotas, Offers and Plans see [Plan-Offer-Quota-Overview](https://docs.microsoft.com/azure/azure-stack/azure-stack-plan-offer-quota-overview)

* A tenant subscribed to your Azure Stack Offer/Plan.

For information on how to Subscribe to an offer see. [Subscribe-to-an-Offer](https://docs.microsoft.com/azure/azure-stack/azure-stack-subscribe-plan-provision-vm)

**Azure**

* An Azure Subscription

If you don't have an Azure subscription, create a [free account](https://azure.microsoft.com/free/?WT.mc_id=A261C142F) before you begin.

* The Azure user needs to have a GitHub Account linked to email
* The Azure user needs to have access to the GitHub Repository

For information on how to add Collaborators see [Adding-Collaborators](https://help.github.com/articles/inviting-collaborators-to-a-personal-repository/)

* You must approve a connection from Azure to GitHub before you begin the deployment. This can be accomplished by manually creating a WebApp from the Azure portal, clicking on the Deployments options and setting up the access to the GitHub repository.

**NOTE:** ASDK Issue: the latest AzureRM commands are not loaded into the Azure Stack SDK environment. So, for this exercise I will copy the Hybrid-Deployment on both my Azure Stack host machine and on a separate machine that has the latest Azure RM commands from the web Platform Installer.

* 1. Before you begin

Verify that you have met the following criteria before beginning your configuration:

* Verify that you have an externally facing public IPv4 address for your VPN device. This IP address cannot run through network address translation (NAT).
* Ensure all resources are deployed in the same region/location.

For more information about VPN Gateway settings in general, see [About VPN Gateway Settings](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-gateway-settings).

**Note: If you are using an ASDK environment please Complete** [**Appendix Section 8.1**](#Appendix8_1)

1. Deploying Azure Stack Resources

In This section you will provision all the necessary resources required to create a Site-to-Site connection between Azure Stack and Azure. The resources deployed are as follows:

* VM’s (All associated resources i.e. NIC’s, Public IP Addresses, V-Net, etc.)
* Network Security Groups
* Standard Storage Account
* Local Network Gateway
* Virtual Network Gateway
* Connection
  1. Preparing Parameters (Azure Stack)

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Download the Hybrid project and save it your local machine |  |
| 2 | Remote into your Azure Stack Environment.  Copy and Paste the **Hybrid Deployment** folder to a location on the Azure Stack VM. |  |
| 3 | So, you will now have two copies | 1- Local Machine  2- Azure Stack Host |
| 4 | On your Azure Stack Host locate the  **Hybrid-Deployment\Hybrid-Azurestack\ azurestackdeploy.parameters.json** file and update the parameters with your settings. | |
| 5 | Fill in **Parameter** **values**. Below is a description of parameters |  |

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Value |
| vmName1 | The Name of your Virtual Machine that will host your SQL Server Instance | Enter a value |
| adminUsername | Local Admin name for your Virtual Machine (Also doubles Sys Admin for SQLDB) | Enter a value |
| adminPassword | Password for Local Admin Account | Enter a value |
| dnsNameForPublicIP | FQDN for Virtual Machine. | Ex: if value is **SQLVM** the DNS Public name will be **SQLVM.local.cloudapp.azurestack.external** |
| AddressPrefix | Virtual Network IP Range | 10.100.100.0/22  If you enter your own values make sure they do not overlap with your Azure Network Range |
| Subnet | Network subnet IP Range (Must be inline with Virtual Network Range) | 10.100.100.0/24 |
| GatewaySubnet | Network subnet for Virtual Network gateway (Must be inline with Virtual Network Range | 10.100.101.0/24 |
| LocalGatewayIPAddress | IP Address of you Azure Gateway Public IP | Leave this value as is. You will not get this value until you deploy your azure resources. |
| LocalGatewayAddressPrefix | The Network IP Address range in your Azure Environment | 10.100.104.0/22 |
| baseURL |  | leave blank as this value gets populated and updated automatically after running script. |

**Note: If you want to deploy Using Azure Command Line Interface Commands please skip to** [**Section 4.3**](#ASCLI1)

* 1. Deploying Template using Azure PowerShell (Azure Stack)

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | In a PowerShell window navigate to the hybrid-Deployment Directory |  |
| 2 | Now run the **Deploy-SolutionAzureStack.ps1** with the following parameters  .\ Deploy-SolutionAzureStack.ps1 -rg “enter value”-storageAccountName “enter value ” – targetStorageContainer “Enter Value” -presharedkey “enter value” | |
| 3 |  | |
| 4 | You will be prompted to enter **Credentials**. This will be the credentials for your **Azure Stack Tenant Subscription** |  |
| 5 | You will get progress output periodically |  |
| 6 | After your Azure Stack Resources have been deployed you will prompted to **Update Local Network Gateway.**  Wait for this value to be outputted when deploying your azure resources. |  |
| 7 | Minimize the PowerShell window |  |

* 1. Deploying Template using Azure CLI Commands (Azure Stack)
     1. Editing Alias.Json file

In this section we need to edit the Alias.json file to correlate with the images you have in your Azure Stack Market Place. The minimum images required are an Image of Windows Server 2016 Datacenter and SQL Server 2016 Enterprise SP1.

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
|  | Edit and or Add values according to images you have in your Azure Stack Market Place  This file can be found in the Hybrid-Deployment root folder  **To Find values from portal see below**  **if you do not have access to admin portal you may want to contact your Azure Stack Admin** |  |
|  | Log into the Azure Stack Admin Portal and Click on **Market Place Management**  Here you will see **Name**, **Publisher** and **Version** values.  Use these values to update your **Alias.json** to represent your **Azure Stack Environment** |  |

* + 1. Create Storage Account for Alias File

In this section we will create a storage account to store our Alias.json file. This is needed when deploying resources to Azure Stack using Azure Command Line Interface.

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
|  | From the Azure Stack Portal click **+New** and Search for **Storage Account** |  |
|  | Enter a Unique name for storage Account **(must be in lower case letters)**  Choose  **Local or Geo Redundant (Depends on your needs)**  Resource Group  Chose the **Create New** and enter a **Name**  Then click **Create** |  |
|  | Once the storage account is created click on the **storage account name** then click on **Blobs** |  |
|  | Click **+Container**  Enter Name  Public access level  **Container**  Then Click **OK** |  |
|  | Click on the **Container** and then upload **alias**.**json** file |  |
|  | Once uploaded copy the value of the **URL** as we are going to need this in a later step |  |

* + 1. Deployment

In this section we will export an Azure Stack certificate and bind it to the Azure stack SDK. Once this is done the script starts the deployment process of Azure Stack Resources

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
|  | In a PowerShell window navigate to the hybrid-Deployment Directory | |
| 1 |  | |
| 2 | Now run the **CLI-AzureStack.ps1** with the following parameters  .\ CLI-AzureStack.ps1 -rg “enter value”-storageAccountName “enter value ” – targetStorageContainer “Enter Value” -certFolderName “Enter Value” -aliasURL “”-presharedkey “enter value” | |
| 3 |  | |
| 4 | You will be prompted to enter **Credentials**. This will be the credentials for your **Azure Stack Tenant Subscription** |  |
| 6 | After your Azure Stack Resources have been deployed you will prompted to **Update Local Network Gateway.**  Wait for this value to be outputted when deploying your azure resources. |  |
| 7 | Minimize the PowerShell window |  |

1. Deploying Azure Resources

As of the writing of this document the location parameters need to be hard coded in the arm template as AppServer V2 environment.  **The App Service management infrastructure isn't correctly handling the normalized location string returned from the resourceGroup.Location() call**. The current azuredeploy file has the location westus2 hardcoded. If you are deploying resources in any other region you will need to change hardcoded values.

* 1. Preparing Parameters (Azure)

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Copy the **test.csv** file that is generated located in the root of the **Hybrid-Deployment** folder and paste it into the **Hybrid-Deployment** folder on your **local machine** |  |
| 2 | Navigate to **Hybrid-Deployment\Hybrid-Azure** folder as we need to edit the **Azuredeploy.paramaters.json file** |  |
| 3 | Edit the Highlighted values. Below is a table with the **parameter** descriptions. |  |

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Value |
| AddressPrefix | Virtual Network IP Range | 10.100.104.0/22  If you enter your own values make sure they do not overlap with your Azure Stack Network Range |
| Subnet | Network subnet IP Range (Must be inline with Virtual Network Range) | 10.100.104.0/24 |
| GatewaySubnet | Network subnet for Virtual Network gateway (Must be inline with Virtual Network Range | 10.100.105.0/24 |
| AppSubnet | Network subnet for your AppServer (Must be inline with Virtual Network Range | 10.100.106.0/24 |
| LocalGatewayIPAddress | External Facing IP Address, cannot be behind NAT | Enter value |
| SiteName | Name of your Website which is hosted on your WebApp Server | Enter value |
| HostingPlanName | Name or your WebApp Server | Enter value |
| EnvrinmentName | Name for the Environment which is Hosting your Isolated Web App Server | Enter value |
| repoURL | This is the URL where the Hybrid Deployment project accessed |  |

**Note: If you want to deploy Using Azure Command Line Interface Commands please skip to** [**Section 5.2**](#ASCLI2)

* 1. Deploying Template using Azure PowerShell (Azure)

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | In a PowerShell window navigate to the hybrid-Deployment Directory |  |
| 2 | Now run the **Deploy-SolutionAzure.ps1** with the following parameters:  .\ Deploy-SolutionAzureStack.ps1 -rg “enter value”-location “enter value “-presharedkey “enter value used earlier” | |
| 3 |  | |
| 4 | You will be prompted to enter **Credentials**. This will be the credentials for your **Azure Subscription** |  |
| 5 | You will get progress output periodically |  |
| 6 | Once Azure resources are deployed you will be prompted to Change Values of Network Gateway |  |
| 7 | **Copy** this value as we are going to use it in a later section |  |

* 1. Deploying Template using Azure CLI (Azure)

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | In a PowerShell window navigate to the hybrid-Deployment Directory |  |
| 2 | Now run the **CLI-Azure.ps1** with the following parameters:  .\ CLI-Azure.ps1 -rg “enter value”-location “enter value “-presharedkey “enter value used earlier” | |
| 3 |  | |
| 4 | You will be prompted to open a browser and enter code |  |
|  | Enter the Code given and click Continue |  |
|  | enter **Credentials**. This will be the credentials for your **Azure Subscription** |  |
| 6 | Once Azure resources are deployed you will be prompted to Change Values of Network Gateway |  |
| 7 | **Copy** this value as we are going to use it in a later section |  |

* + 1. Updating AppSettings.Json File

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | From the Azure Public portal Click on your Web App, Scroll down and Click on **Advanced Tools**, then Click **Go** |  |
| 2 | From Kudu Window Click on **Debug Console**, then on **CMD** |  |
| 3 | Click On **Site,wwwroot,** then the P**encil icon** to the left of the **appsettings.json file** |  |
| 4 | Open the **Appsettings.json** file located in **Hybrid-Deployment** folder then **Copy Contents** |  |
|  | **Paste Contents into KUDO console** andclick **Save** |  |

* 1. Configuring Azure Connection

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Go back to your **Azure Stack host** and log into **tenant portal** using credentials used in your **Azure Stack Deployment** | |
| 2 | Navigate to the resource group created in [Step 2 in Section 4.2](#AzureStack_RG)  Click on the **Local Network Gateway** icon |  |
| 3 | Click on the **Configurations** tab |  |
| 4 | Paste the value of IP address given in [Section 5.2 step 7](#AzurePIP) |  |
| 5 | Navigate back to your PowerShell window in your Azure Stack host and press **Any Key to Continue** |  |
| 6 | Once you see that the **Connection** has **succeeded** you will be returned to the **PowerShell prompt** |  |
| 7 | From your local machine navigate to your PowerShell window and press **Any Key to Continue** |  |
| 8 | Once you see that the **Connection** has **succeeded** you will be returned to the **PowerShell prompt** |  |

1. Verifying Deployment

In this section we will go over the steps needed to verify deployment.

* 1. Azure Connection

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Log into your Azure (Public) portal  **http://Portal.azure.com**  Navigate to the recent created **connection,** you should now see a “**Connected**” status  Below are screen grabs of both Azure and Azure Stack Connections | |
| 2 | Azure Stack View |  |
| 3 | Log into your Azure Stack Tenant portal  Navigate to the recent created **connection,** you should now see a “**Connected**” status | |
| 4 | Azure View |  |

* 1. WebApp Application Settings

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Click on your Web app |  |
| 2 | Click the **Application Settings** Tab and verify that the value of your connection string |  |
| 3 | **Connection String Name** = SQLWM  **Value** = Data Source**={Internal IP Address of SQLVM}**,1433;Initial Catalog=NorthwindDb;User ID**={Name used for adminUsername in Paramters}**;Password**={value of adminPassword in Paramters**}Asynchronous Processing=True  **Type** = SQLServer | |

* 1. WebApp Application files

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Scroll down and Click on **Advanced Tools**, then Click **Go** |  |
| 2 | From Kudu Window Click on **Debug Console**, then on **CMD** |  |
| 3 | Click On **Site,wwwroot,** then the P**encil icon** to the left of the **appsettings.json file** |  |
| 4 | Verify that the Default **connection** **string** matches the **value** displayed in [Step 2 of Section 7.1](#AppSettings) |  |
| 5 | Verify that the **App insights** **value** matches.  You can see the value of **App insights f**rom the **Portal.**  See next step to validate |  |
| 6 | From the Azure Portal **click App Insights Icon**, **Instrumentation Key** is on the top right of window |  |

* 1. WebApp Connectivity

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Next use Tcpping to check connectivity between the WebApp and SQLVM using port 1433  In the lower window type:  Tcpping **10.00.100.4:1433**  **You should get a successful response**  **Note: If you are following guidance using default values enter the above IP Address.**  **If you used other network segments in deployment use the value of SQLIP in the test.csv file located in the Root folder of Hybrid-Deployment with a :1433** |  |

1. Appendix
   1. Configuring BGP for Azure Stack Development Kit Only

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Log in to the Azure Stack physical host for your ASDK |  |
| 2 | From the Hyper-V Manager Console  Click Az**S-BGPNAT01**  On the lower window click on Networking tab  Take note of the IP address for the NAT Adapter that uses the PublicSwitch |  |
| 3 | Right Click Az**S-BGPNAT01** and click **Connect** button |  |
| 4 | Sign into VM |  |
| 5 | Verify that IP’s match by typing ipconfig /all  One of your IP addresses should match to the value from the Networking Tab in previous step  In my environment it’s **10.16.169.131**  but yours will be something different |  |
| 6 | In the command prompt type  **Start PowerShell**  And press **Enter** |  |
| 7 | Enter the PowerShell command  To designate the external NAT address for the ports that the IKE Authentication tunnel will use.  Remember to change the IP Address to the value seen in | Add-NetNatExternalAddress -NatName BGPNAT -IPAddress 10.16.169.131 -PortStart 499 -PortEnd 501 |
| 8 | Enter the PowerShell command  Create a static NAT mapping to map the external address to the Gateway Public IP Address.    This maps the ISAKMP port 500 for PHASE 1 of the IPSEC tunnel | Add-NetNatStaticMapping -NatName BGPNAT -Protocol UDP -ExternalIPAddress 10.16.169.131 -InternalIPAddress 192.168.102.1 -ExternalPort 500 -InternalPort 500 |
| 9 | Finally, we will need to do NAT traversal which uses port 4500 to successfully establish the complete IPSEC tunnel over NAT devices | Add-NetNatStaticMapping -NatName BGPNAT -Protocol UDP -ExternalIPAddress 10.16.169.131 -InternalIPAddress 192.168.102.1 -ExternalPort 4500 -InternalPort 4500 |
| 10 | If you run a Get-NetNatExternalAddress -Natname BGPNAT  You should see similar results | Get-NetNatExternalAddress -Natname BGPNAT |

* 1. Using the North Wind WebApp

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Navigate to your website, on the landing page Fill out dummy information and clik the **Show Plans** Button |  |
| 2 | On the **Plans page** choose an option by clicking **Buy Now** |  |
| 3 | On the **Apply for a Plan** Page fill out dummy information and click **Submit Appli****cation** |  |
| 4 | You should now receive a **Confirmation Code** |  |

* 1. Verify Data in Database

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | From your Azure Stack host remote to your SQLVM |  |
| 2 | Open SQL Server Management Studio |  |
| 3 | Log into your SQL Server, Expand  Northwinddb, Tables, Right-Click and chose “Select Top 1000 rows” |  |
| 4 | The result of the Query should show the Person created in [Step 3 Section 8.1](#Webapp) | |
| 5 |  | |