Azure Stack to Azure Hybrid Connection with WebApp

Technical guidance

Contents

[1 Overview 2](#_Toc524132531)

[1.1 Context and Considerations 2](#_Toc524132532)

[1.2 When to Use This Pattern 2](#_Toc524132533)

[2 Reference Architecture 3](#_Toc524132534)

[2.1 Azure Resources 3](#_Toc524132535)

[2.2 Azure Stack Resources 4](#_Toc524132536)

[3 Prerequisites 5](#_Toc524132537)

[3.1 Before you begin 5](#_Toc524132538)

[4 Deploying Azure Stack Resources 6](#_Toc524132539)

[4.1 Preparing Parameters (Azure Stack) 6](#_Toc524132540)

[4.2 Deploying Template (Azure Stack) 8](#_Toc524132541)

[5 Deploying Azure Resources 9](#_Toc524132542)

[5.1 Preparing Parameters (Azure) 9](#_Toc524132543)

[5.2 Deploying Template (Azure) 11](#_Toc524132544)

[6 Configuring VPN Connection 12](#_Toc524132545)

[6.1 Azure Stack Connection Configuration 12](#_Toc524132546)

[6.2 Azure Connection Configuration 14](#_Toc524132547)

[7 Configuring WebApp for VNET Routing 15](#_Toc524132548)

[7.1 Configuring Point to Site Connection 16](#_Toc524132549)

[7.2 Configuring Vnet Integration for WebApp 17](#_Toc524132550)

[7.3 Syncing Routes and Cert for Appserver 19](#_Toc524132551)

[8 Troubleshooting 24](#_Toc524132552)

[8.1 Verifying Azure VPN tunnel 24](#_Toc524132553)

[8.2 Verifying WebApp Application Settings 25](#_Toc524132554)

[8.3 Verifying Appsettings.Json file 26](#_Toc524132555)

[8.4 WebApp Connectivity 28](#_Toc524132556)

[9 Appendix 29](#_Toc524132557)

[9.1 Configuring BGP for Azure Stack Development Kit Only 29](#_Toc524132558)

[9.2 Using the North Wind WebApp 33](#_Toc524132559)

[9.3 Verify Data in Database 35](#_Toc524132560)

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. 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.
  1. Before you begin

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

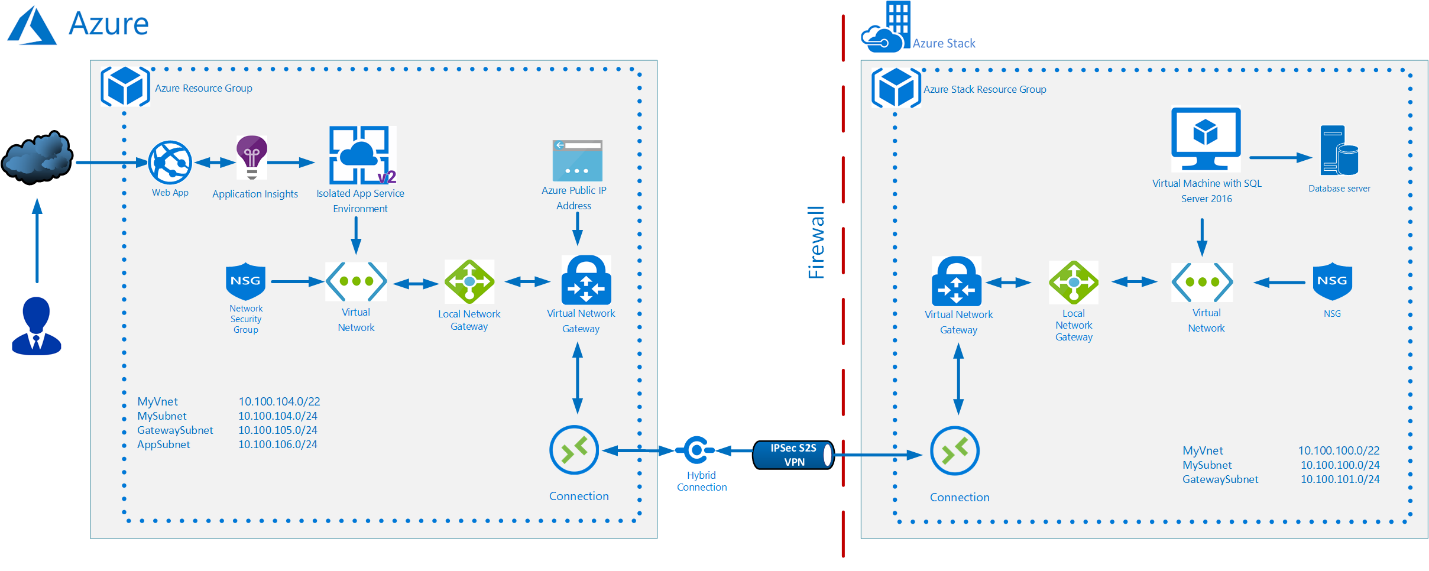
* Verify that you have an externally facing public IPv4 address for your VPN device. This IP address cannot run through network address translation (NAT).
* Ensure all resources are deployed in the same region/location.

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 9.1**

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. 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 |  |
| 5 | Navigate to the **Azurestackdeploy.paramaters.json** file  It is located in the **Hybrid-AzureStack** folder |  |
|  | Fill in **Parameter** **values**. Below is a description of parameters |  |

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Value |
| dnsNameForPublicIP | FQDN for Virtual Machine. | Ener a Value |
| 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. |

* 1. Deploying Template (Azure Stack)

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Open a **PowerShell ISE** window as an **Administrator** and navigate to the **hybrid-Deployment Directory** | |
|  |  | |
| 2 | Now run the **Deploy-SolutionAzureStack.ps1** with the following parameters  .\Deploy-SolutionAzureStack.ps1 -rg {enterValue} -presharedkey {enterValue} -storageAccountName {enterValue} -targetStorageContainer {enterValue} -location {enterValue} -tenantID {enterValue} -azureStackArmEndpoint {enterValue} -Verbose  .\Deploy-SolutionAzureStack.ps1 -rg {enterValue} -presharedkey {enterValue} -storageAccountName {enterValue} -targetStorageContainer {enterValue} -AADTenantName {enterValue} -azureStackArmEndpoint {enterValue} -Verbose | |
| 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 be prompted to **Update Local Network Gateway.**  **At this point please Minimize the PowerShell ISE window.** |  |
|  | **We need to get IP value from section 5.2 before we resume deployment** | |

1. Deploying Azure Resources

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

* WebApp
* App Insights
* Network Security Groups
* Local Network Gateway
* Virtual Network Gateway
* Connection
  1. Preparing Parameters (Azure)

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 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 |
| 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 | entervalue |

* 1. Deploying Template (Azure)

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Open a PowerShell window as Administrator and navigate to the **hybrid-Deployment Directory** |  |
| 2 | Now run the **Deploy-SolutionAzure.ps1** with the following parameters:  .\Deploy-SolutionAzureStack.ps1 -rg {enterValue} - -presharedkey {enterValue} -ADAppPassword {enterValue} -emailNotification {enterValue}  -Verbose | |
|  | **Be sure to use the same preshared key value used earlier** | |
| 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 your Local Network Gateway in Azure Stack  **Copy** this value as we are going to use it in a later section  **At this point please Minimize the PowerShell window.** |  |

1. Configuring VPN Connection

In this Section we will configure our Local Network Gateways with the outputted values given from both the PowerShell ISE and PowerShell windows.

* 1. Azure Stack Connection Configuration

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Log into your **Azure Stack Portal** | |
| 2 | Navigate to the resource group created in [Step 2 in Section 4.2](#Resourcegroup_4_2)  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 6](#IP_5_2)  And Click **Save** |  |
| 5 | Navigate back to your **PowerShell ISE window**  **Be sure that it is the ISE Window that is associated to your Azure Stack**  **Press** **Any Key to Continue** |  |
| 6 | Once you see that the **Connection** has **succeeded**  Copy the Value as we will need this for the next section |  |

* 1. Azure Connection Configuration

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Log into **Azure Public Portal** | |
| 2 | Navigate to the resource group created in [Step 2 in Section 5.2](#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 6.1 Step 6](#IP_6_1)  And Click **Save** |  |
|  | After pressing Enter you should see a **provisioning State Succeeded** you have now completed the VPN Connection between **Azure and Azure Stack** |  |

1. Configuring WebApp for VNET Routing

A common scenario where you would use VNet Integration is enabling access from your web app to a database or a web service running on a virtual machine in your Azure virtual network. With VNet Integration, you don't need to expose a public endpoint for applications on your VM but can use the private non-internet routable addresses instead.

The VNet Integration feature:

* Requires a Standard, Premium, or Isolated pricing plan
* Works with Classic or Resource Manager VNet
* Supports TCP and UDP
* Works with Web, Mobile, API apps and Function apps
* Enables an app to connect to only 1 VNet at a time
* Enables up to five VNets to be integrated with in an App Service Plan
* Allows the same VNet to be used by multiple apps in an App Service Plan
* Supports a 99.9% SLA due to the SLA on the VNet Gateway

There are some things that VNet Integration does not support including:

* mounting a drive
* AD integration
* NetBios
* private site access

Here are some things to keep in mind before connecting your web app to a virtual network:

* VNet Integration only works with apps in a **Standard**, **Premium**, or **Isolated** pricing plan. If you enable the feature, and then scale your App Service Plan to an unsupported pricing plan your apps lose their connections to the VNets they are using.
* If your target virtual network already exists, it must have point-to-site VPN enabled with a Dynamic routing gateway before it can be connected to an app. If your gateway is configured with Static routing, you cannot enable point-to-site Virtual Private Network (VPN).
* The VNet must be in the same subscription as your App Service Plan(ASP).
* If your gateway already exists with point-to-site enabled, and it is not in the basic SKU, IKEV2 must be disabled in your point-to-site configuration.
* The apps that integrate with a VNet use the DNS that is specified for that VNet.
* By default your integrating apps only route traffic into your VNet based on the routes that are defined in your VNet.
  1. Configuring Point to Site Connection

In this section we will need to configure our routes between our Azure and Azure Stack Networks. This is done by configuring the VNET Routing.

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Log into your Azure Subscription. Then navigate to your Virtual Network Gateway. |  |
|  | Click on **Point-to-Site Configuration**. |  |
|  | Click on **Configure Now** |  |
|  | Under Address Pool. Enter **172.16.0.0/24**    **Be sure that SSL VPN is checked**  And Click **Save** |  |

* 1. Configuring VNET Integration for WebApp

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Navigate to your resource group window and Click on your **WebApp** |  |
|  | Scroll down and Click on **Network** |  |
|  | Click **Setup** |  |
|  | Chose **myVnet** |  |
|  | Getting a Failure is expected. If you refresh the page. You will see that your **VNET integration** is in a **Connected** state  See picture Below. |  |
|  | View of page after refresh |  |

* 1. Syncing Routes and Cert for Appserver

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Navigate to your resource group window and Click on your **AppServer** |  |
|  | Scroll down and Click on **Network** |  |
|  | Click **Manage** |  |
|  | Click on **myVnet** |  |
|  | Next Click on Sync Network. Then click **Yes** |  |
|  | A Cmak error is expected here.  This happens because of the IKEv2 and External Radius Settings configured in [Section 7.1](#Vnet_7_1) |  |
|  | **In your PowerShell Window (the one Connected to Azure public)**  **Type**  **.\cmak\_error\_fix.ps1** | |
|  |  | |
|  | Enter Azure Credentials |  |
|  | Once the script is finished running, Go back to your **Azure Public Portal**. |  |
|  | Clear all **notifications** |  |
|  | Now try **Syncing Network** again |  |
|  | Click **Yes** |  |
|  | You should now see 3 green Check Marks stating that **Certificates, routes and Data** have initialized a sync. You have now successfully configured routes between your **Azure and Azure Stack** Networks. |  |

* 1. Add DNS Host Name to Azure Stack Web App

|  |  |  |
| --- | --- | --- |
| Step | Step Details | Screen Capture |
| 1 | Navigate to your Azure Stack portal and click on WebApp |  |
| 2 | Click on Custom Domains Tab |  |
| 3 | Click on Add HostName |  |
| 4 | Paste value outputted in the Powersehll window with traffic manager endpoint |  |
| 5 | Click on Add Host Name |  |
| 6 | Then in the PowerShell Window hit enter to continue |  |

* 1. Getting Key Secrets

|  |  |  |
| --- | --- | --- |
| 1 | Click on Azure Active Directory |  |
| 2 | Click on App Registration |  |
| 3 | Click on Add HostName |  |
| 4 | Locate the “Trigger” app and click on it  Note: You might have to switch drop down box to All Apps |  |
| 5 | Click On Keys |  |
| 6 | Enter a Name under Description change expiration to 2 years and Click Save |  |
| 7 | Copy And Paste the Secret given to you into the Power Shell window  BE Sure not to Close Window without Copying as you will not see this Secret Again |  |
| 8 | Once you Pasted Secret in PowerShell press enter |  |

* 1. Upload Files to Azure Function

|  |  |  |
| --- | --- | --- |
| 1 | Navigate to your Function Web App |  |
| 2 | Click on Http Trigger |  |
| 3 | On the Right Column click on View Files |  |
| 4 | Here we need to replace the files in this App with the file located in  .\cross-cloud-scale\httpTrigger |  |
| 5 | Upload One By One overwriting files |  |

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  **values are taken from**  **Change 10.16.169.131 to your IP Address** | Add-NetNatExternalAddress -NatName BGPNAT -IPAddress 10.16.169.131 -PortStart 499 -PortEnd 501  Add-NetNatExternalAddress -NatName BGPNAT -IPAddress 10.16.169.131 -PortStart 1500 -PortEnd 1501 |
| 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  **Change 10.16.169.131 to your IP Address** | Add-NetNatStaticMapping -NatName BGPNAT -Protocol UDP -ExternalIPAddress 10.16.169.131 -InternalIPAddress 192.168.102.1 -ExternalPort 500 -InternalPort 5 |
| 9 | Finally, we will need to do NAT traversal which uses port 4500 to successfully establish the complete IPSEC tunnel over NAT devices  **Change 10.16.169.131 to your IP Address** | 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 click 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 Application** |  |
| 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 8.2](#sql_8_2) | |
| 5 |  | |

1. Troubleshooting
   1. Verifying Azure VPN tunnel

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

|  |  |  |
| --- | --- | --- |
| 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. Verifying 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. Verifying Appsettings.Json file

|  |  |  |
| --- | --- | --- |
| 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** |  |