

# AZURE FOUNDATIONS FOR INFRASTRUCTURE

## OVERVIEW

Welcome to Azure Foundations for Infrastructure. This design pattern represents the culmination of hundreds of customer cloud deployments throughout the past few years. From these deployments, several key patterns emerged, and many best practices were created. The design patterns described here includes all these elements in a simple to understand manner and includes an application to help you get started. The application is named *Foundations Editor* and is available here:

<https://github.com/csajeflan/FoundationsEditor/blob/master/FoundationsEditor.zip>

The application is designed to gather the minimum amount of information to create and deploy your base network infrastructure for IaaS deployments in Microsoft Azure. A key design goal of this pattern is to allow for a modular approach to creating your cloud environment. It begins small and can be expanded as your cloud usage increases. The basic pattern begins with two subscriptions in your Server Cloud Enrollment: One for production services and one for pre-production services.

## CONTENTS

Overview .....	1
Azure Enrollment and Subscription Management .....	3
Subscription Management Patterns .....	4
Services Based Hub and Spoke Pattern .....	4
Production Pre-Production Pattern .....	5
Departmental Subscription Pattern .....	5
Administrative Access and Permissions Management .....	6
Departmental Subscriptions .....	6
Role Based Access Controls .....	6
Resource Tagging .....	7
Network Connectivity and Hybrid Networking .....	7
Network Address Space and Subnetting .....	8
Subnet Definitions .....	8
Standard Naming Conventions .....	10
Network Naming Convention .....	10
Type-Subtype-Location-Additional Elements .....	10
Resource Group Naming Convention .....	10
Resource Group Identifier-Type-Subtype-Location .....	10
Customization and Deployment .....	10
Using the Foundations Editor .....	11
Field Entries and Selections .....	11
Foundations Editor Generated Files .....	12
Sample PowerShell Script Generated Output .....	13
Sample ARM Template Generated Output .....	14
Post Deployment Objects – Azure Portal .....	14
Connections and Local Gateways .....	14
VNET-to-VNET Connection .....	14
Network Security Groups .....	15
DMZ NSG .....	15
Virtual Networks .....	16
Subnets .....	16

## AZURE ENROLLMENT AND SUBSCRIPTION MANAGEMENT

An Azure Server Cloud Enrollment is the highest level of abstraction and is the primary container for all Azure components. An enrollment always contains at least one subscription but can contain many more if required. There are also departments and accounts that are managed at the enrollment level. Governance is a key concept for determining your specific best practice in organizing and implementing your Azure environment. There are many other links and documents that describe governance in greater detail. A good place to start is this link: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-subscription-governance>.

Using service accounts to administer enrollments and subscriptions is a recommended best practice. A single user account should not be assigned as the primary account owner. Instead, create a dedicated service account for this purpose, as well as using a distribution group for the email address. This ensures that you will always have the ability to access these within your organization as it is not tied to one particular individual.

The following graphic depicts the hierarchy of an Azure enrollment. A short article that describes the relationships between enrollments, departments, and accounts is available here: <https://marckean.com/2016/06/03/azure-enterprise-enrollment-hierarchy/>. Thanks to Marc Kean for providing this information.

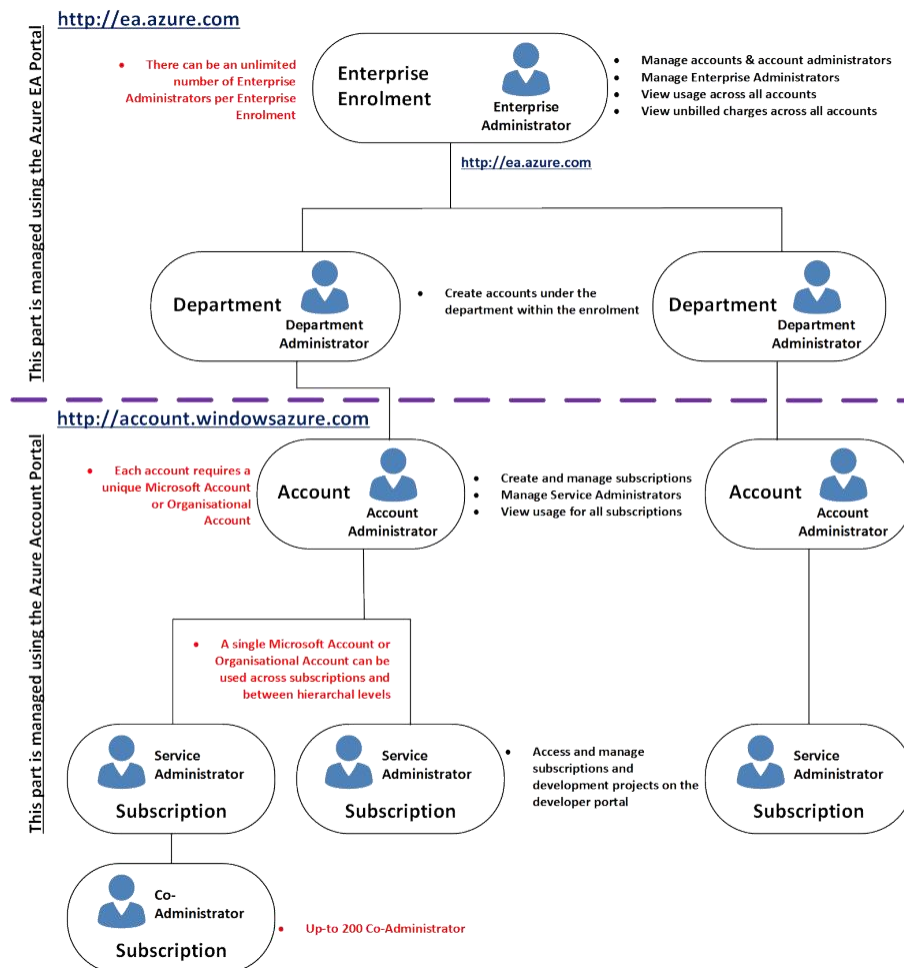


Figure 1 - Enrollment Hierarchy

## SUBSCRIPTION MANAGEMENT PATTERNS

The organization and determination of your specific subscription needs will ultimately be based on the governance that you decide. We will discuss three common patterns here for your consideration.

### SERVICES BASED HUB AND SPOKE PATTERN

This pattern consists of five (5) subscriptions:

- |                   |  |
|-------------------|--|
| 1. SERVICES       | Central services such as networking, identity, etc.                                      |
| 2. HBI            | High Business Impact. Workloads that require specific compliance or security separation. |
| 3. STORAGE        | Archival or Historical storage that requires additional security or separation.          |
| 4. PRODUCTION     | Production workloads.  |
| 5. PRE-PRODUCTION | Development and testing workloads.   |

All centralized or shared IT services are contained in the SERVICES subscription. Other workloads are deployed depending on their specific requirements. Networking between all subscriptions and on-premises is contained in this subscription.

The diagram below depicts a fully H/A environment for all five subscriptions deployed in two separate Azure regions.

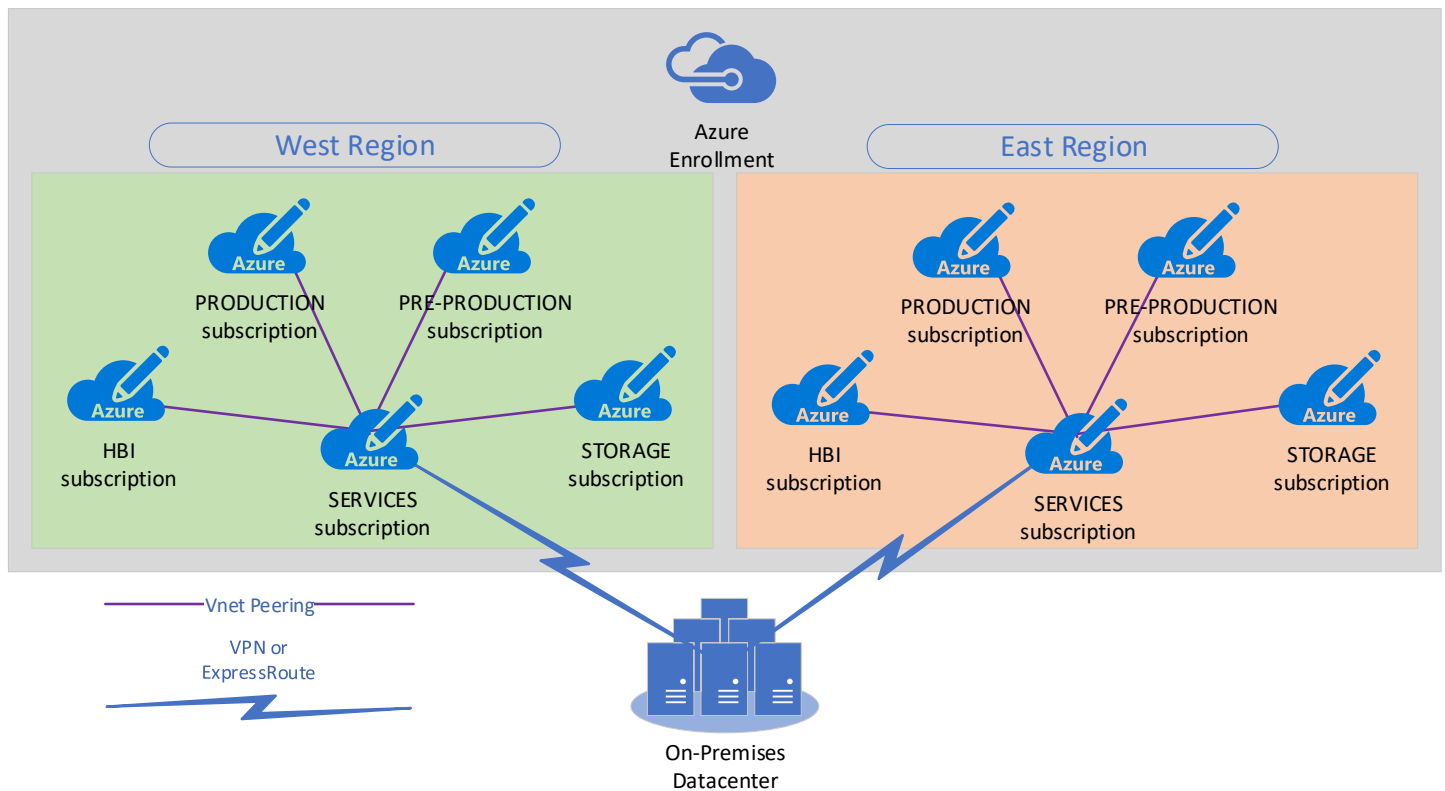


Figure 2 - Services Based Hub and Spoke Pattern

## PRODUCTION PRE-PRODUCTION PATTERN

This pattern consists of only two (2) subscriptions. One for production workloads and one for non-production workloads. This allows for separation of duties and permissions between IT staff and development staff. With this pattern, only IT staff would have administrative rights in the production subscription, while development staff would be free to do what they need autonomously in the pre-production subscription. However, many administrative functions such as networking can be locked down with this pattern so that the development staff could deploy services that they needed, but they could not change the networking configuration.

The diagram below depicts this pattern. Only the production subscription is highly available (H/A) in this model, although the pre-production subscription can also be added if desired.

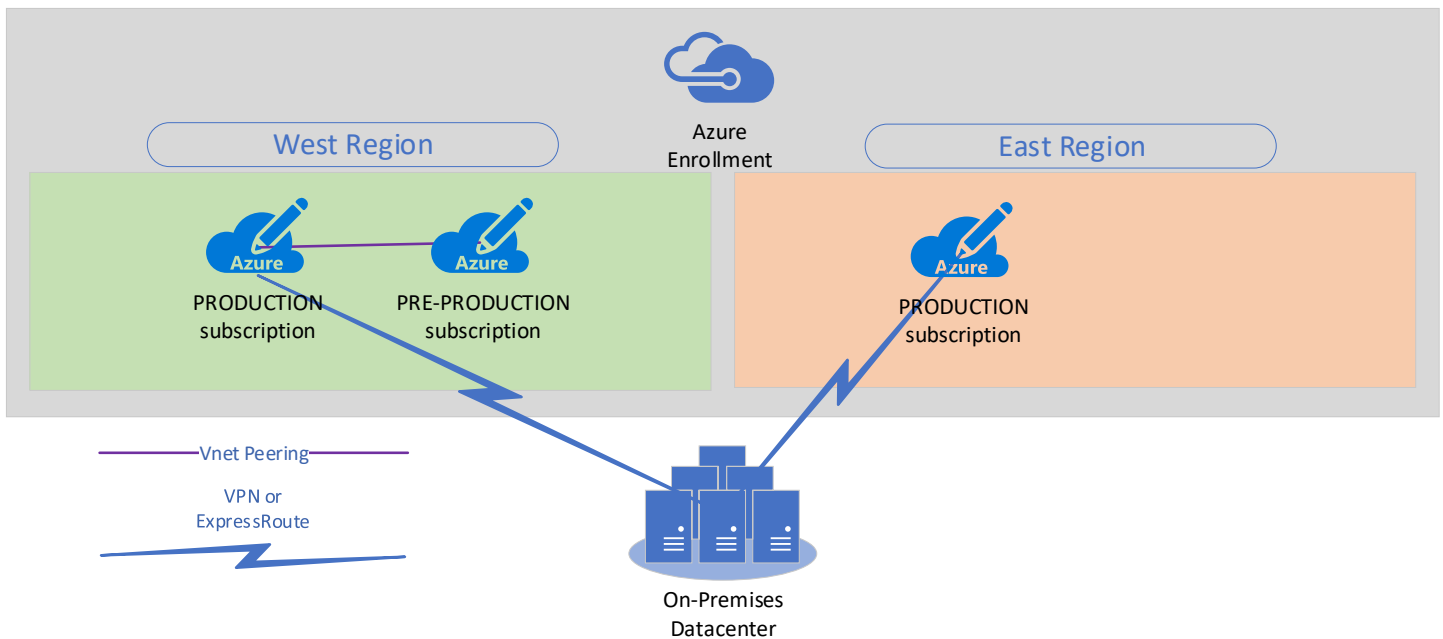


Figure 3 - Production Pre-Production Pattern

## DEPARTMENTAL SUBSCRIPTION PATTERN

The departmental subscription pattern is not unlike that of the services hub and spoke pattern. However, it is designed to give flexibility to individual departments—especially those departments that have their own IT staff. This is typically seen in larger enterprise or government organizations. One benefit to this pattern is that an individual subscription can be moved out of the enrollment into another enrollment if that is desired or necessary. This is a combination of the production pre-production and services pattern in that each department would typically have two subscriptions: one for production workloads and one for development and testing workloads.

The diagram on the next page depicts what a typical departmental subscription pattern looks like. Initially, all departments are created in a central State, County or City enrollment. If it is determined later that the Department of Public Safety workloads must be separated and administered independently from the larger organization because of budgetary or compliance requirements, these individual subscriptions can be easily moved from one enrollment to another.

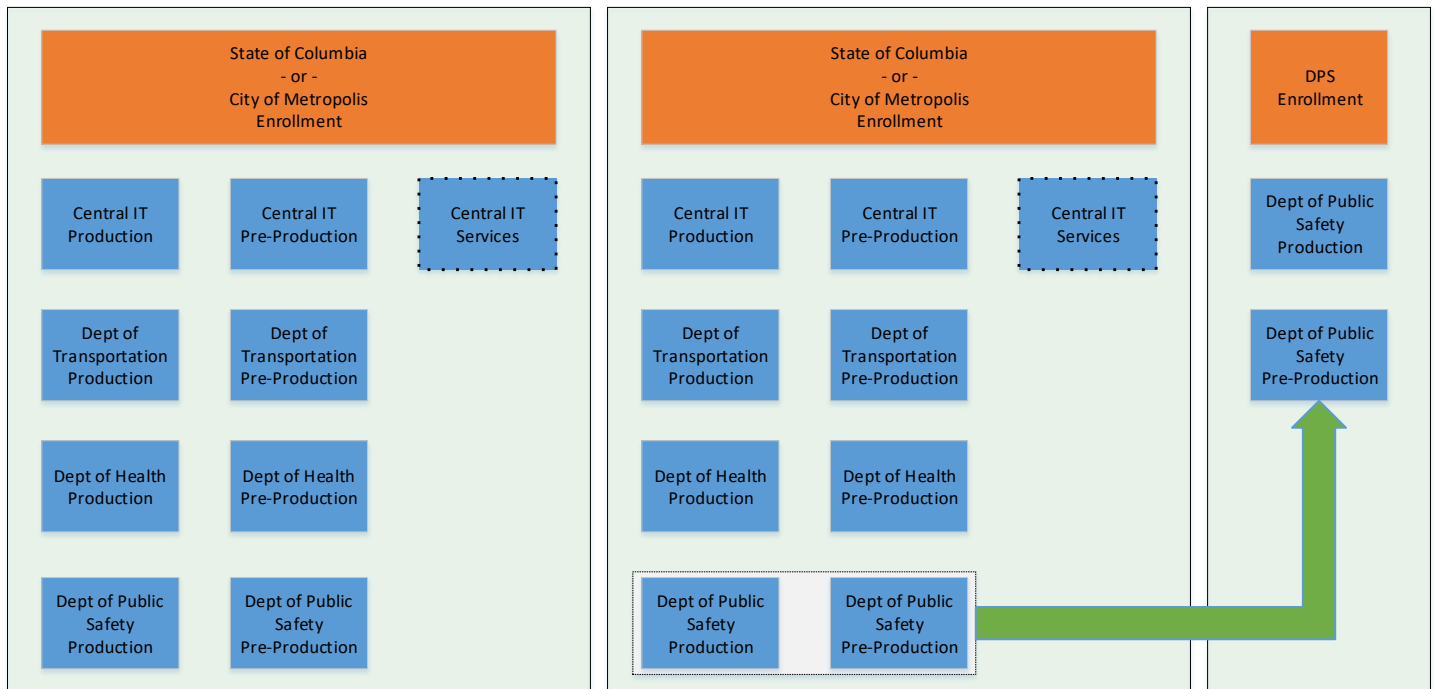


Figure 4 - Departmental Subscription Pattern

## ADMINISTRATIVE ACCESS AND PERMISSIONS MANAGEMENT

Controlling user access and permissions is necessary in any organization. Separation of workloads and users by department or function is often a requirement of any governance structure. There are a few different ways to accomplish this, and often a combination of these will be used.

### DEPARTMENTAL SUBSCRIPTIONS

The first mechanism to consider is the use of Subscriptions for larger departments that have their own dedicated IT staff. An Azure Subscription is an administrative and logical boundary. One Subscription does not know about any other Subscription and is not connected in any way to other Subscriptions within an enrollment—with two exceptions: The only two things shared between Subscriptions are the Azure Active Directory namespace and the billing. All Subscriptions within an enrollment share the same monetary commit bucket of funds. As such, there is a single repository for funding all cloud services within an organization.

Individual users or groups are granted access to a Subscription on an as-required basis. A central IT staff may have rights in multiple Subscriptions, while a departmental user may only have access to a single Subscription. This limits what any given user can do within the enterprise environment. If a given department is primarily self-sufficient in their cloud IT needs, specified users can be given global admin rights for that subscription which enables them to do anything that is required within it. This is especially true in the Production Pre-Production governance model. Developers can be granted a much higher level of access to a Pre-Production Subscription, with no direct access to the Production Subscription. Typical end users of application workloads that are deployed in Azure do not require any direct access to Azure. They will simply be users of the applications and therefore do not need to manage any Azure objects directly.

### ROLE BASED ACCESS CONTROLS

RBAC allows the partitioning of rights within a desired scope. These are typically applied at the Resource Group level. Individual permissions such as networking, virtual machine creation, specific PaaS services access, etc. can be granted to users or groups as required. This allows for a much more granular level of access than the Subscription level global admin role. The RBAC mechanism can delegate permissions to specific workloads based on the role of an end user. More information about RBAC is located here:

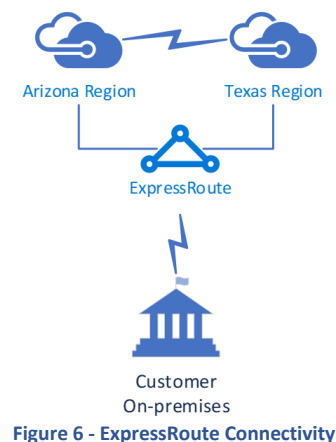
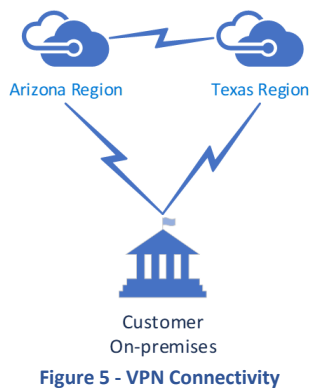
## RESOURCE TAGGING

Resource Tagging allows every resource in Azure to be identified in some way and can be utilized for billing purposes as well as general organization of workloads by department or function. Typically, a tag is added to every resource that is created in Azure. They can be applied by department, application, function (such as database, front-end, application, etc.). You can apply multiple tags to any resource, which allows for billing separation as well as application or function separation. More information about tagging is located here: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-using-tags#tagging-with-resource-manager>

## NETWORK CONNECTIVITY AND HYBRID NETWORKING

The Production Pre-Production pattern is the simplest one to start with. As previously mentioned, it is easy to add more subscriptions to your environment if it is desired to do so. With that in mind, all the networking configurations described in this section will be using that pattern.

The pattern creates Site-to-Site VPNs between the on-premises location and one or two Azure regions. A third Azure-to-Azure VPN is created between the two regions if both a primary and secondary region is defined. Connections between the Production and Pre-Production subscription virtual networks is provided by the utilization of network peering so that the number of VPNs is kept to a minimum. It is recommended to start with VPN and then move to ExpressRoute once it is available—especially since it typically takes several weeks to procure and implement an ExpressRoute circuit.



## NETWORK ADDRESS SPACE AND SUBNETTING

The virtual network address space is initially created as a /20 or /21 which is adequate for most small to medium sized enterprises. This can be increased to a larger size as required, but keep in mind that maximum number of allowed IP addresses in any one virtual network is limited to a /20 (4096 addresses). The virtual networks are further divided into subnets--each with a specific purpose. It is highly recommended that the minimum number of subnets you should create is eight (8). This section will address why that is necessary.

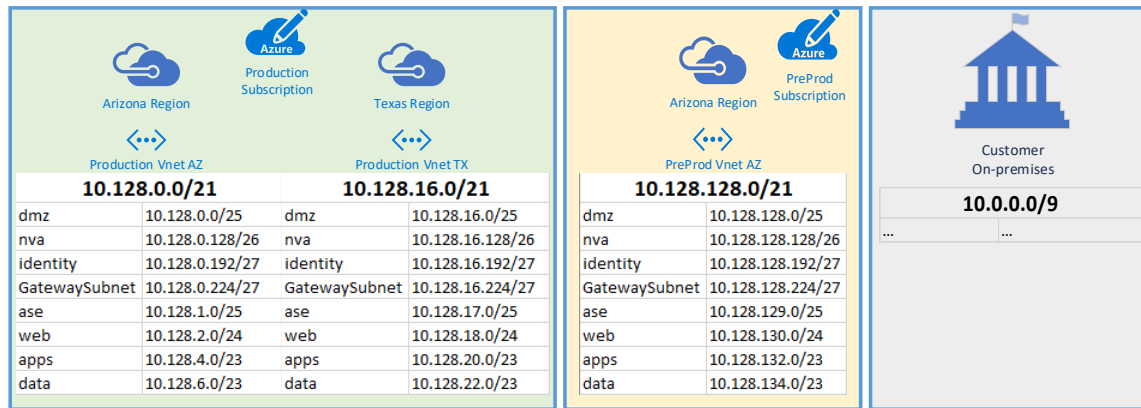


Figure 7 - Subnet Definitions

### SUBNET DEFINITIONS

#### DMZ

The DMZ subnet is designed for all Internet facing front-ends. It can be secured independently from other subnets by using a Network Security Group (NSG). If it is desired to use a Network Virtual Appliance (NVA) instead of NSGs, a User Defined Route (UDR) would be applied to route all traffic to the NVA subnet. Direct inbound Internet access is allowed for this subnet.

#### NVA

The NVA subnet is designed for use by Network Virtual Appliances. In most cases, this will be a third-party NVA to provide a firewall and specific routing configurations. This subnet can also be utilized for an Application Gateway if desired. Direct inbound Internet access is allowed for this subnet.

#### IDENTITY

The Identity subnet is designed for use by active directory domain controllers, ADFS, MIM, and other identity related servers. ADFS web proxy servers would be deployed in the DMZ subnet as they are Internet facing. Direct inbound Internet access is blocked for this subnet (NSG or NVA).

#### GATEWAYSUBNET

The gateway subnet is required for both VPN and ExpressRoute gateways. This subnet must be a /27 or larger and is typically created at the end of the first /24 segment.

#### ASE

The ASE subnet is designed to hold the App Services Environment (ASE). This is a dedicated subnet for an internally facing or Vnet connected ASE. It can also contain a Public IP (PIP) in Azure if desired. This subnet must be at least a /25 in size.

#### WEB



The web subnet is designed to hold the private internal only facing front-end servers. Direct inbound Internet access is blocked for this subnet (NSG or NVA).

---

## APPS

The apps subnet is designed for middle-tier application servers. These are typically accessed from either the DMZ or Web subnets. These servers generally handle the communication to the database or backend servers as well, so both inbound and outbound connectivity to the data subnet is enabled. Direct inbound Internet access is blocked for this subnet (NSG or NVA).

---

## DATA

The data subnet contains database and other backend process servers. Direct inbound Internet access is blocked for this subnet (NSG or NVA).

---

## SUBNET ALLOCATION TABLE

10.128.0.0/21		Production Vnet AZ
dmz	10.128.0.0/25	Public facing web servers
nva	10.128.0.128/26	Network Virtual Appliances
identity	10.128.0.192/27	Identity layer servers
GatewaySubnet	10.128.0.224/27	VPN E/R Gateway
ase	10.182.1.0/25	App Services Environment
web	10.128.2.0/23	Internal facing web servers
apps	10.128.4.0/23	Application (middle tier) servers
data	10.128.6.0/23	Database and backend servers
10.128.16.0/21		Production Vnet TX
dmz	10.128.16.0/25	Public facing web servers
nva	10.128.16.128/26	Network Virtual Appliances
identity	10.128.16.192/27	Identity layer servers
GatewaySubnet	10.128.16.224/27	VPN E/R Gateway
ase	10.128.17.0/25	App Services Environment
web	10.128.18.0/23	Internal facing web servers
apps	10.128.20.0/23	Application (middle tier) servers
data	10.128.22.0/23	Database and backend servers
10.128.128.0/21		Pre-production Vnet AZ
dmz	10.128.128.0/25	Public facing web servers
nva	10.128.128.128/26	Network Virtual Appliances
identity	10.128.128.192/27	Identity layer servers
GatewaySubnet	10.128.128.224/27	VPN E/R Gateway
ase	10.128.129.0/25	App Services Environment
web	10.128.130.0/23	Internal facing web servers
apps	10.128.132.0/23	Application (middle tier) servers
data	10.128.134.0/23	Database and backend servers

## STANDARD NAMING CONVENTIONS

The naming conventions utilized in this pattern are based on the blog post here:

<https://blogs.msdn.microsoft.com/azuregov/2017/03/29/microsoft-azure-iaas-architecture-best-practices-for-arm/>. The portions highlighted indicate components that are automatically generated with the use of the *Foundations Editor*. For instance, the application prompts for the name of the virtual networks in each region. You would enter only **vnet-prod-az** as the name for the primary virtual network, and the PowerShell script that it generates would automatically append the name with the additional elements.

## NETWORK NAMING CONVENTION

### TYPE-SUBTYPE-LOCATION-ADDITIONAL ELEMENTS

vnet-prod-usgovarizona	Denotes Virtual Network, Production, USGovArizona
vnet-prod-usgovarizona-gw	Denotes Virtual Network Gateway
vnet-prod-usgovarizona-gw-ip	Denotes Virtual Network Gateway Public IP Address
gw-local-usgovarizona	Denotes the local gateway for the AZ region
gw-local-usgovtexas	Denotes the local gateway for the TX region
cn-local-usgovarizona	Denotes the connection between on-premises and the AZ region
cn-vnet-prod-usgovarizona-vnet-prod-usgovtexas	Denotes the connection between the Azure regions

## RESOURCE GROUP NAMING CONVENTION

### RESOURCE GROUP IDENTIFIER-TYPE-SUBTYPE-LOCATION

rg-vnet-prod-usgovarizona	Denotes all the networking components for the USGovArizona region
rg-vnet-prod-usgovtx	Denotes all the networking components for the USGovTexas region

## CUSTOMIZATION AND DEPLOYMENT

Customizing a deployment for you to use is designed to be a simple and straight forward exercise that requires very little effort. The *Foundations Editor* application has been designed to gather the specific required information and can then be used to automatically generate the necessary PowerShell script and ARM templates to deploy the pattern to your subscription(s). The *Foundations Editor* will automatically calculate the subnetting for a /21 or /20 address space. As such, you need only enter the base IP address space in the PRIMARY IP Segment text field (such as 10.128.0.0/21), and it will fill in the rest after you check the [Automatic IP Range](#) checkbox.

The application only automatically calculates the CIDR segmentation for a /21 or /20 size because using a size smaller than /21 is not generally recommended, and using a size larger than /20 gives more available private IP addresses than a single virtual network can utilize. Larger address spaces can still be entered, however they must be calculated and entered manually, which creates the segmentation you desire. The ASE subnet is created as a /25, which leaves another /25 available for a second ASE, or another function as desired. There will be a gap of the same size as configured between the primary and secondary locations. For instance, if you specify a /21, the primary region will use 10.xxx.0.0 – 10.xxx.7.255, and the secondary region will use 10.xxx.16.0 – 10.xxx.23.255. This leaves another /21 between the two unused for future expansion if required.

There are only eight fields for subnets, which is the minimum recommended. If more than eight subnets are desired, they can be added via the portal or PowerShell after the initial creation. The order of the subnets in the application is important: this is how the NSGs get automatically created and applied. It is assumed that the *GatewaySubnet* is the first field, followed by the *DMZ* and *NVA*. The rest of the CIDR calculations are determined by the workloads intended to be deployed to the subnets. You can change the actual names if desired (except for the *GatewaySubnet*), but the functionality of the workloads remain the same. You may also manually edit the CIDR ranges after automatic calculation if you want them to be different.

*Foundations Editor* generates a PowerShell script to deploy the virtual network(s), gateway(s), and connection(s) as desired. If you only want to deploy the virtual network and not the gateway(s) or connection(s), you simply leave those checkboxes unchecked. If further customization is desired, both the PowerShell script and the ARM templates can be manually edited prior to deployment.

The *Foundations Editor* application is modular by design, and works at the individual subscription level. You have the option of deploying a highly available pattern to two separate Azure regions, or to just a single region (such as for a pre-production only subscription). If you utilize more than one subscription in your final environment design, you will run the application once for each subscription. Connections between subscriptions are not in the scope of the application, and those must be completed manually via the portal or PowerShell after all of the individual subscription virtual networks have been deployed.

## USING THE FOUNDATIONS EDITOR

The Editor is distributed in zip file that contains a folder with several files in it. You need only double-click on the *FoundationsEditor.exe* file to open it. The two JSON files in the folder contain the predefined subnet names and Azure locations. The locations included in this file only contain the US based regions for both MAC and MAG. The DoD regions are not included but could be added if desired. The same is true for regions located outside of the US. The proper region pair will be automatically selected when you choose the primary region. You can still opt to deploy it to any other region if desired by selecting a different one from the secondary dropdown list.

app.publish	3/1/2018 10:35	File folder	
FoundationsEditor.application	3/1/2018 10:35	Application Manif...	2 KB
FoundationsEditor.exe	3/1/2018 10:35	Application	443 KB
FoundationsEditor.exe.config	2/9/2018 10:19	XML Configuratio...	1 KB
FoundationsEditor.exe.manifest	3/1/2018 10:35	MANIFEST File	7 KB
FoundationsEditor.pdb	3/1/2018 10:35	Program Debug D...	72 KB
locations.json	2/26/2018 10:59	JSON Source File	1 KB
Newtonsoft.Json.dll	2/18/2018 09:44	Application extens...	649 KB
Newtonsoft.Json.xml	2/18/2018 09:44	XML Document	669 KB
subnets.json	2/28/2018 15:50	JSON Source File	1 KB
System.Management.Automation.dll	9/28/2017 22:29	Application extens...	1,576 KB

Figure 8 - Foundations Editor Files

Figure 9 - Foundations Editor Application

## FIELD ENTRIES AND SELECTIONS

1. Enter the desired Azure Subscription ID. This field expects a valid GUID.
2. Select either **MAC** or **MAG** in the **Environment** panel. MAC=Microsoft Azure Commercial, MAG=Microsoft Azure Government
3. Select either **Primary Only** or **Both** in the **Region Deployment** panel. Primary Only=deploy only a single region, Both=deploy to both a Primary and Secondary region.
4. Choose the desired primary location from the dropdown list in the **PRIMARY** column. These values change depending on the **Environment** selection. The secondary location will be automatically selected based on its region pair if **Both** is selected in the **Region Deployment** panel.

5. If **Both** is selected and you wish to specify a location that is different from the primary region pair, choose the desired secondary location from the dropdown list in the SECONDARY column. Otherwise skip to step 6. These values change depending on the [Environment](#) selection. Choose a location that is different from the Primary.
6. Enter the desired Resource Group name for the PRIMARY column. Follow the desired naming convention (e.g. rg-vnet-prod-az).
7. Enter the desired Resource Group name for the SECONDARY column. Follow the desired naming convention (e.g. rg-vnet-prod-tx).
8. Enter the desired Virtual Network name for the PRIMARY column. Follow the desired naming convention (e.g. vnet-prod-az).
9. Enter the desired Virtual Network name for the SECONDARY column. Follow the desired naming convention (e.g. vnet-prod-tx).
10. Enter the desired IP range for the PRIMARY column. This range must not overlap with the SECONDARY or On-Premises IP ranges.  
**NOTE: Skip to step 12 if this range is a /21 or /20 in size**
11. Enter the desired IP range for the SECONDARY column. This range must not overlap with the PRIMARY IP range.  
**NOTE: Skip to step 13 if your IP range is not a /21 or /20 in size**
12. Click on the Automatic IP Range checkbox. All other subnets are automatically calculated and filled in—including the SECONDARY IP range if Both is selected in the [Region Deployment](#) panel.
13. If desired, you can change the subnet names except for the *GatewaySubnet* in the first subnet row.
14. If desired or if your IP range is not a /21 or /20 in size, you can manually change the subnet IP ranges and sizes, however no error checking is provided for this. You will need to ensure that your ranges are valid and do not overlap.
15. If you want the local and VPN gateways to be automatically created in this deployment, click on the Create Gateway(s) checkbox. This will add the gateway creation to the generated PowerShell scripts.
16. If you want the VPN connections to be automatically created, click on the Create Connection(s) checkbox. This will add the S2S and Vnet-to-Vnet connections to the generated PowerShell script.  
**NOTE: The following text fields are only available if you select the Create Gateway(s) checkbox**
17. Enter the desired local gateway name. Follow the desired naming convention (e.g. gw-local).
18. Enter the on-premises edge IP. This is the public IP address of your VPN endpoint that the Azure gateways will connect to.
19. Enter your on-premises IP range. This range cannot overlap with any of the Azure IP ranges. An example would be 10.0.0.0/9. Only enter a single range here. Additional ranges can be added via the Azure Portal after the initial deployment of the Foundations pattern.
20. Verify all fields. If everything looks good, click on SAVE. This will save a JSON file with the configuration settings of this deployment, as well as two separate PowerShell script.
21. Enter the desired filename for this deployment. Do NOT add an extension to this filename! The proper file extensions will be automatically added to this filename by the Foundations Editor application.
22. Close the application. If you want to later edit this deployment, the JSON file that is created when you saved the configuration is what you would LOAD.

## FOUNDATIONS EDITOR GENERATED FILES

Example: The filename entered during the SAVE process was '**deployment**'.

deployment-foundationseditor.json	The configuration file that contains all settings and options selected
deployment-deployARM.ps1	The PowerShell script to deploy the virtual network(s)
deployment-primarynsg.json	The ARM template for the primary region NSGs
deployment-primaryvnet.json	The ARM template for the primary region Vnet
deployment-secondarynsg.json	The ARM template for the secondary region NSGs
deployment-secondaryvnet.json	The ARM template for the secondary region Vnet

## SAMPLE POWERSHELL SCRIPT GENERATED OUTPUT

This is only a small snippet of the actual PowerShell script that is generated automatically.

```
1 #####
2 # Clear Screen and Logon to Azure
3 #####
4 Clear-Host
5 Add-AzureRmAccount
6 $global:script_error = $false
7 Write-Host '#####'
8 Write-Host ' Azure Foundations Deployment Script'
9 Write-Host '#####'
10 Write-Host
11 #####
12 # Select Desired Azure Subscription
13 #####
14 Write-Host
15 Read-Host '*** Press any key to continue ***' | Out-Null
16 Write-Host 'Selecting Subscription: ' -NoNewLine
17 $sub=Get-AzureRmSubscription -SubscriptionID [REDACTED] -ErrorAction Ignore
18 if($sub)
19 {
20     Select-AzureRmSubscription -SubscriptionObject $sub | Out-Null
21     Write-Host 'SUCCESS' -ForegroundColor Green
22 }
23 else
24 {
25     Write-Host 'FAILED- Unable to select subscription' -ForegroundColor Red
26     $global:script_error = $true
27 }
28 #####
29 # Get or Create Primary Location Resource Group
30 #####
31 if($global:script_error -eq $false)
32 {
33     Write-Host 'Checking Primary Location Resource Group: ' -NoNewLine
34     $prg=Get-AzureRmResourceGroup -Name rg-network-prod-westus2 -ErrorAction Ignore
35 }
36 if ($prg) {Write-Host 'WARNING: Resource Group Already Exists--Skipping Creation'}
37 else
38 {
39     Write-Host 'Resource Group Does Not Exist'
40     Write-Host 'Creating Primary Location Resource Group: ' -NoNewLine
41     $prg=New-AzureRmResourceGroup -Name rg-network-prod-westus2 -Location westus2
42     if ($prg) {Write-Host 'SUCCESS' -ForegroundColor Green}
43     else
44     {
45         Write-Host 'FAILED- Unable to create resource group' -ForegroundColor Red
46         $global:script_error = $true
47     }
48 }
```

Figure 10 - PowerShell Script Snippet

```







"$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
"contentVersion": "1.0.0.0",
"parameters": { },
"variables": {
  "vnetID": "[resourceId('Microsoft.Network/virtualNetworks','test-vnet-prod-az')]",
  "gatewaySubnetRef": "[concat(variables('vnetID'), '/subnets/', 'GatewaySubnet')]"
},
"resources": [
  {
    "name": "gw-local-usgovarizona",
    "type": "Microsoft.Network/localNetworkGateways",
    "apiVersion": "2017-10-01",
    "location": "[resourceGroup().location]",
    "properties": {
      "localNetworkAddressSpace": {
        "addressPrefixes": [
          "10.0.0.0/9"
        ]
      },
      "gatewayIpAddress": "61.72.83.94"
    }
  },
  {
    "name": "test-vnet-prod-az",
    "type": "Microsoft.Network/virtualNetworks",
    "apiVersion": "2017-10-01",
    "location": "[resourceGroup().location]",
    "properties": {
      "addressSpace": {
        "addressPrefixes": [
          "10.192.0.0/21"
        ]
      },
      "subnets": [
        {
          "name": "GatewaySubnet",
          "properties": {
            "addressPrefix": "10.192.0.224/27"
          }
        },
        {
          "name": "dmz",
          "properties": {
            "addressPrefix": "10.192.0.0/25"
          }
        }
      ]
    }
  }
]
```

Figure 11 - ARM Template Snippet

POST DEPLOYMENT OBJECTS – AZURE PORTAL

After running the script that was generated by the *Foundations Editor*, a look into the Azure portal will show you everything that has been deployed. **NOTE:** the objects shown here are from a deployment to Azure Commercial rather than the previous section which was created for Azure Government using a /21 virtual network size. The deployment shown in this section was created using a /20 virtual network size which allows for larger subnets.

CONNECTIONS AND LOCAL GATEWAYS

NAME	TYPE	RESOURCE GROUP	RESOURCE TYPE	LOCATION
 cn-local-eastus2	Connection	rg-network-prod-eastus2	Microsoft.Network/connections	East US 2
 cn-local-westus2	Connection	rg-network-prod-westus2	Microsoft.Network/connections	West US 2
 cn-vnet-prod-eastus2-vnet-prod-westus2	Connection	rg-network-prod-eastus2	Microsoft.Network/connections	East US 2
 cn-vnet-prod-westus2-vnet-prod-eastus2	Connection	rg-network-prod-westus2	Microsoft.Network/connections	West US 2
 gw-local-eastus2	Local network gateway	rg-network-prod-eastus2	Microsoft.Network/localNetworkGateways	East US 2
 gw-local-westus2	Local network gateway	rg-network-prod-westus2	Microsoft.Network/localNetworkGateways	West US 2

VNET-TO-VNET CONNECTION

cn-vnet-prod-westus2-vnet-prod-eastus2  
Connection

Overview

Activity log

Access control (IAM)

Tags

SETTINGS

Shared key

Configuration

Properties

Locks

Automation script

SUPPORT + TROUBLESHOOTING

Resource health

New support request

→ Move

🗑 Delete

Resource group [\(change\)](#)

rg-network-prod-westus2

Status

Connected

Location

West US 2

Subscription [\(change\)](#)

Microsoft Azure Commercial - JEFLAN

Subscription ID

Data in

13.38 KiB

Data out

13.41 KiB

Virtual network

vnet-prod-eastus2, vnet-prod-westus2

Virtual network gateway 1

vnet-prod-westus2-gw (52.151.11.178)

Virtual network gateway 2

vnet-prod-eastus2-gw (52.177.191.219)

## NETWORK SECURITY GROUPS

nsg-apps-eastus2	Network security group	rg-network-prod-eastus2	Microsoft.Network/networkSecurityGroups	East US 2
nsg-apps-westus2	Network security group	rg-network-prod-westus2	Microsoft.Network/networkSecurityGroups	West US 2
nsg-ase-eastus2	Network security group	rg-network-prod-eastus2	Microsoft.Network/networkSecurityGroups	East US 2
nsg-ase-westus2	Network security group	rg-network-prod-westus2	Microsoft.Network/networkSecurityGroups	West US 2
nsg-data-eastus2	Network security group	rg-network-prod-eastus2	Microsoft.Network/networkSecurityGroups	East US 2
nsg-data-westus2	Network security group	rg-network-prod-westus2	Microsoft.Network/networkSecurityGroups	West US 2
nsg-dmz-eastus2	Network security group	rg-network-prod-eastus2	Microsoft.Network/networkSecurityGroups	East US 2
nsg-dmz-westus2	Network security group	rg-network-prod-westus2	Microsoft.Network/networkSecurityGroups	West US 2
nsg-identity-eastus2	Network security group	rg-network-prod-eastus2	Microsoft.Network/networkSecurityGroups	East US 2
nsg-identity-westus2	Network security group	rg-network-prod-westus2	Microsoft.Network/networkSecurityGroups	West US 2
nsg-nva-eastus2	Network security group	rg-network-prod-eastus2	Microsoft.Network/networkSecurityGroups	East US 2
nsg-nva-westus2	Network security group	rg-network-prod-westus2	Microsoft.Network/networkSecurityGroups	West US 2
nsg-web-eastus2	Network security group	rg-network-prod-eastus2	Microsoft.Network/networkSecurityGroups	East US 2
nsg-web-westus2	Network security group	rg-network-prod-westus2	Microsoft.Network/networkSecurityGroups	West US 2

## DMZ NSG

Azure Foundations for Infrastructure

[Document Version 1.9]

Page 15 of 17

nsg-dmz-westus2  
Network security group

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

SETTINGS

Inbound security rules

Outbound security rules

Network interfaces

Subnets

Properties

Locks

Automation script

MONITORING

Diagnostics logs

SUPPORT + TROUBLESHOOTING

Effective security rules

New support request

Move

Delete

Resource group (change)

rg-network-prod-westus2

Location

West US 2

Subscription (change)

Microsoft Azure Commercial - JEFLAN

Subscription ID

Security rules

3 inbound, 0 outbound

Associated with

1 subnets, 0 network interfaces

Inbound security rules

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
100	Port_80_HTTP	80	Any	Internet	Any	✔ Allow
101	Port_443_HTTPS	443	Any	Internet	Any	✔ Allow
999	⚠ Port_3389_RDP	3389	Any	Internet	Any	✔ Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✔ Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	✔ Allow
65500	DenyAllInBound	Any	Any	Any	Any	✖ Deny

Outbound security rules

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	✔ Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	✔ Allow
65500	DenyAllOutBound	Any	Any	Any	Any	✖ Deny

## VIRTUAL NETWORKS

vnet-prod-eastus2	Virtual network	<a href="#">rg-network-prod-eastus2</a>	Microsoft.Network/virtualNetworks	East US 2
vnet-prod-eastus2-gw	Virtual network gateway	<a href="#">rg-network-prod-eastus2</a>	Microsoft.Network/virtualNetworkGateways	East US 2
vnet-prod-eastus2-gw-ip	Public IP address	<a href="#">rg-network-prod-eastus2</a>	Microsoft.Network/publicIPAddresses	East US 2
vnet-prod-westus2	Virtual network	<a href="#">rg-network-prod-westus2</a>	Microsoft.Network/virtualNetworks	West US 2
vnet-prod-westus2-gw	Virtual network gateway	<a href="#">rg-network-prod-westus2</a>	Microsoft.Network/virtualNetworkGateways	West US 2
vnet-prod-westus2-gw-ip	Public IP address	<a href="#">rg-network-prod-westus2</a>	Microsoft.Network/publicIPAddresses	West US 2

## SUBNETS



<>

vnet-prod-westus2 - Subnets

Virtual network

Search (Ctrl+/)

<<

+ Subnet

+ Gateway subnet

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

SETTINGS

Address space

Connected devices

Subnets

DNS servers

Peerings

Service endpoints

Properties

Locks

Automation script

MONITORING

Diagram

SUPPORT + TROUBLESHOOTING

New support request

Search subnets

NAME	ADDRESS RANGE	AVAILABLE ADDRESSES	SECURITY GROUP
GatewaySubnet	10.128.0.0/25	122	-
dmz	10.128.1.0/24	251	nsg-dmz-westus2
nva	10.128.2.0/25	123	nsg-nva-westus2
identity	10.128.2.128/25	123	nsg-identity-westus2
ase	10.128.3.0/25	123	nsg-ase-westus2
web	10.128.4.0/22	1019	nsg-web-westus2
apps	10.128.8.0/22	1019	nsg-apps-westus2
data	10.128.12.0/22	1019	nsg-data-westus2