Deployment Guide for Azure Autoscaling in Azure Transit VNet for PA-VM Series firewalls

Table of Contents

[Overview 2](#_Toc516484196)

[Pre-requisites for deploying the hub template 3](#_Toc516484197)

[Panorama Server 3](#_Toc516484198)

[Panorama Setup 3](#_Toc516484199)

[Enabling XML API access in Panorama 3](#_Toc516484200)

[Generate Panorama API Key 3](#_Toc516484201)

[Generate Panorama VM Auth Key 3](#_Toc516484202)

[Adding the Licensing Deactivation Key to Panorama 3](#_Toc516484203)

[Azure Setup 4](#_Toc516484204)

[Creating a Service Principal 4](#_Toc516484205)

[Retrieve Tenant ID 4](#_Toc516484206)

[Launching the hub template 5](#_Toc516484207)

[Pre-requisites for launching the Spoke template 6](#_Toc516484208)

[Device Group and Template in Panorama 6](#_Toc516484209)

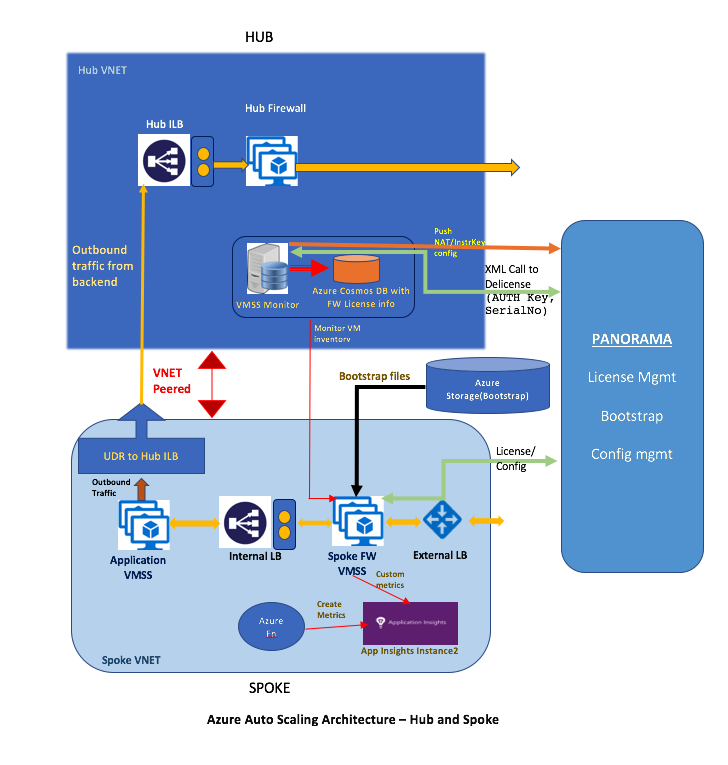
[Bootstrapping package 6](#_Toc516484210)

[Creating the bootstrapping package 7](#_Toc516484211)

[Launching the spoke template 7](#_Toc516484212)

# Overview

The Azure Transit VNet with the VM-Series deploys a hub and spoke architecture to centralize commonly used services such as security and secure connectivity. All traffic to and from the Spokes will “transit” the Hub VNet and will be protected by the VM-Series next generation firewall. More details about these template can be found here - <https://github.com/PaloAltoNetworks/Azure-Transit-VNet>.

However, the Azure Transit VNET did not have auto scaling enabled and hence had a static number of PA-VMs deployed. Autoscaling helps in optimizing resources by scaling in/out based on demand. The following diagram gives an overview of the deployment.

As mentioned above, the deployment consists of two templates – a hub and spoke(s). A hub is where consists of firewalls handling all the outbound traffic originating from the application backend. It also contains a worker VM which monitors the spokes subscribed to this hub and manages the PA-VMs launched in the spoke through Panorama.

# Pre-requisites for deploying the hub template

## Panorama Server

We depend on a working Panorama server to manage licensing for BYOL licenses and hence a reachable Panorama server with internet connectivity is a pre-requisite to deploy the template. The following are the setup steps for Panorama

### Panorama Setup

How to setup Panorama is outside the purview of this document. Guidance on how to setup a Panorama VM can be found here - <https://www.paloaltonetworks.com/documentation/71/panorama/panorama_adminguide/set-up-panorama/set-up-the-panorama-virtual-appliance#55656>

We need at least Panorama 8.1 for this feature.

Output of this step would be a reachable Panorama IP. Let us call this input as **PanoramaIP** for future references.

### Enabling XML API access in Panorama

In order to program Panorama and to deactivate VM Licenses, we need to enable XML API access in Panorama. As a best practice, create a new role with just API access to perform this. Steps to do this can be found here - <https://www.paloaltonetworks.com/documentation/71/pan-os/xml-api/get-started-with-the-pan-os-xml-api/enable-api-access>

### Generate Panorama API Key

To authenticate the API, we need a Panorama API Key. The following link helps in generating an API Key for the user created in the previous step. <https://www.paloaltonetworks.com/documentation/71/pan-os/xml-api/get-started-with-the-pan-os-xml-api/get-your-api-key>

Let us call this input as **PanoramaAPIKey** for future references.

### Generate Panorama VM Auth Key

Since we want to use Panorama to manage the VM-Series firewalls that we are bootstrapping, we must generate a VM auth key on Panorama and include the key in the basic configuration (init-cfg.txt) file. Steps to generate the VM Auth key can be found here - <https://www.paloaltonetworks.com/documentation/71/virtualization/virtualization/bootstrap-the-vm-series-firewall/generate-the-vm-auth-key-on-panorama>

Let us call this input as **PanoramaVmAuthKey** for future references.

### Adding the Licensing Deactivation Key to Panorama

We require a License Deactivation API Key and the “Verify Update Server Identity” to be enabled in order to deactivate the license keys from Panorama. The License Deactivation Key should be obtained from Palo Alto Customer Support Portal. Steps on how to activate this can be found at - <https://www.paloaltonetworks.com/documentation/71/virtualization/virtualization/license-the-vm-series-firewall/install-a-license-deactivation-api-key>

## Azure Setup

We need the following information to enable API Access in Azure

* Subscription Id
* Azure Application ID
* Azure Application Secret Key
* Tenant id

### Creating a Service Principal

To enable API access for Azure resources, we would need to create an Azure App and Service Principal and associate them together. We need to associate “Contributor” role to the App. Details on how to perform this step can be found here - <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal>. The output of this step will be the Azure Application Id (**AzureAppId**) and Azure Application Secret Key (**AzureAppSecret**). To associate roles, you have to be “Owner” or “User Access Administrator” role in Azure.

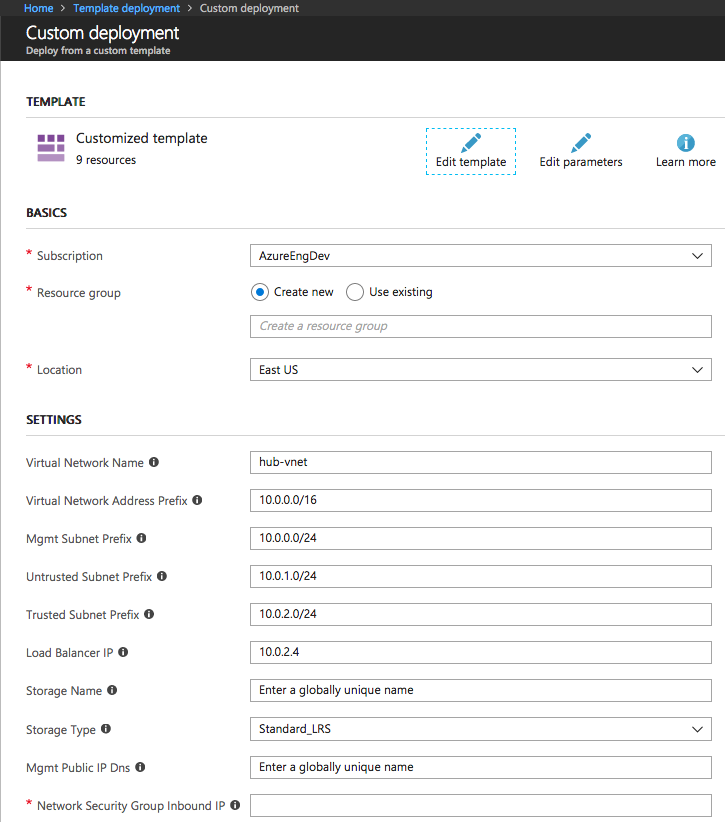
### Retrieve Tenant ID

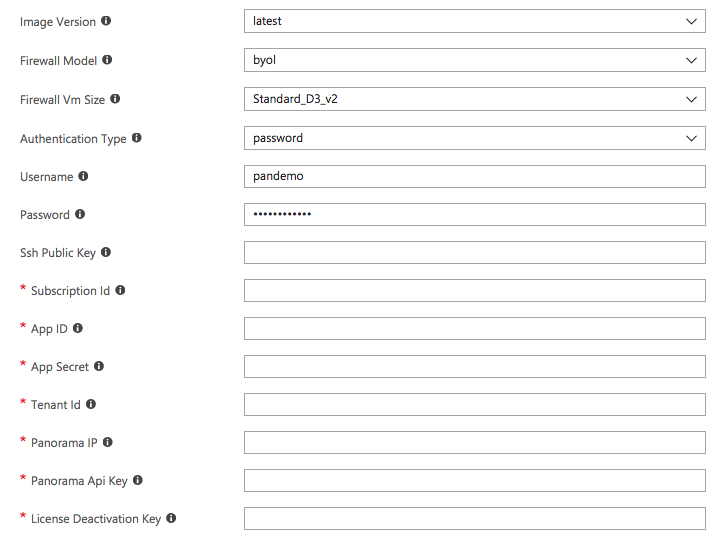
We can retrieve the tenant-id as follows. It is also called as Directory Id in Azure.

<https://stackoverflow.com/questions/26384034/how-to-get-the-azure-account-tenant-id>. Let us call this **AzureTenantId** for future references.

# Launching the hub template

Use the parameters obtained above to fill up the hub template. A sample hub config (working-hub-config.xml) is enclosed to bootstrap the hub PA-VMs (Bootstrapping of the hub PA-VMs is not supported). The template is currently hosted at <https://github.com/PaloAltoNetworks/azure-autoscaling/tree/master/phase2/azure-hub/azureDeployInfra.json>.



The hub’s virtual network name and the load Balancer IP are inputs to the spoke template as well. The pre-requisites described above will help you on how to obtain the input values for the fields described above.

# Pre-requisites for launching the Spoke template

## Device Group and Template in Panorama

For every spoke that is launched, a corresponding Device Group, Template and Template Stack needs to be created in Panorama.

The DG should be named as <spoke\_name> + “-dg” so that it can be identified by the hub monitoring script. For example, if the spoke name is spoke-1, then the DG name in Panorama has to be spoke-1-dg.

The Template Stack should be named as <spoke\_name> + “-tmplstk” so that it can be identified by the hub monitoring script. For example, if the spoke name is spoke-1, then the DG name in Panorama has to be spoke-1-tmplstk.

The Device Group should also have a dummy object called ILB\_NAT\_ADDR created with a random IP address which will re-programmed by the hub’s monitoring script.

## Bootstrapping package

It is recommended that we create a separate resource group with storage for hosting the bootstrapping files since we can re-use the storage for multiple spokes.

### Creating the bootstrapping package

The following link provide details on how to create a bootstrapping package in general.

<https://www.paloaltonetworks.com/documentation/71/virtualization/virtualization/bootstrap-the-vm-series-firewall/prepare-the-bootstrap-package#_38054>. A sample bootstrap package is provided as bootstrap.tar.gz as part of the template repo.

A sample init-cfg.txt is provided below with explanation –

type=dhcp-client

ip-address=

default-gateway=

netmask=

ipv6-address=

ipv6-default-gateway=

hostname=

vm-auth-key=**PanoramaVmAuthKey**

panorama-server=**PanoramaIP**

panorama-server-2=

tplname=<spoke\_name> + ”-tmplstk”

dgname=<spoke\_name> + ”-dg”

dns-primary=8.8.8.8

dns-secondary=208.67.222.222

op-command-modes=

dhcp-send-hostname=yes

dhcp-send-client-id=yes

dhcp-accept-server-hostname=yes

dhcp-accept-server-domain=yes

There are a few naming conventions to be followed in Panorama while creating the device group and template group for a spoke.

# Launching the spoke template

Use the parameters obtained above to fill up the spoke template. The template is currently hosted at <https://github.com/PaloAltoNetworks/azure-autoscaling/tree/master/phase2/azure-spoke/azureDeploy.json>.

