Pulse Policy Secure Virtual Appliance on Microsoft Azure

Deployment Guide

Document Revision	1.1
Published Date	May 2020

Pulse Secure, LLC
2700 Zanker Road,
Suite 200 San Jose
CA 95134

https://www.pulsesecure.net

Pulse Policy Secure Virtual Appliance on Microsoft Azure Cloud - Deployment Guide

The information in this document is current as of the date on the title page.

END USER LICENSE AGREEMENT

Revision History

Revision and Date	Added/Updated /Removed	Remarks
April 2020	Pulse Policy Secure on Azure Market Place is newly added.	
January 2020	First Version for 9.1R4.	

Table of Contents

Overview

About This Guide

This guide helps in deploying the Pulse Policy Secure Virtual Appliance on Microsoft Azure. The PPS 9.1R4 image is now available in Azure Marketplace.
This document also describes how a Pulse Policy Secure administrator manually upload the Pulse Policy Secure Virtual Appliance image into Microsoft Azure storage account. Once the image is available in the Azure storage account, this document describes how the Pulse Policy Secure administrator can deploy Pulse Policy Secure on Microsoft Azure.

Assumptions

The basic understanding of deployment models of Pulse Policy Secure on a data center and basic experience in using Microsoft Azure is needed for the better understanding of this guide.

Pulse Policy Secure on Azure Marketplace

Prerequisites and System Requirements on Azure Marketplace

To deploy the Pulse Policy Secure Virtual Appliance on Azure Marketplace, you need the following:

- A Microsoft Azure account
- Access to the Microsoft Azure portal (https://portal.azure.com)
- Pulse Policy Secure licenses *

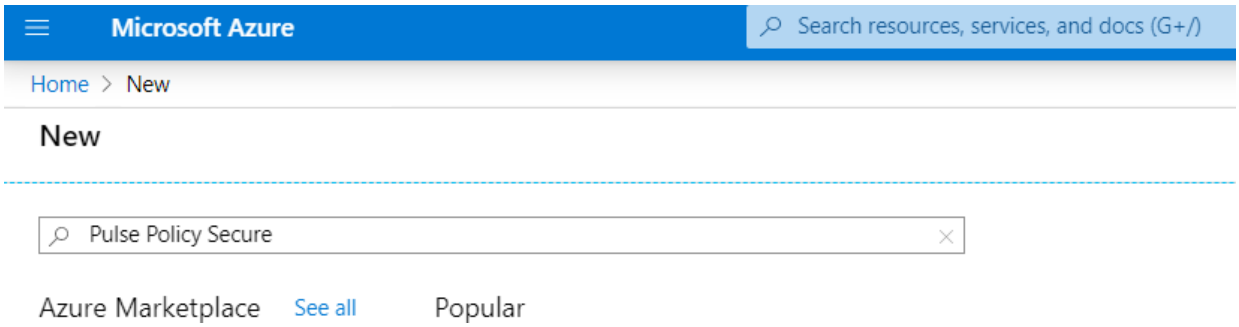
Note:
* Pulse Policy Secure Virtual Appliance, by default, has two-users license. This release supports licensing with License server located at corporate network and licensing through Pulse Cloud Licensing Service (PCLS) server. For licensing through PCLS, administrator needs to obtain Authentication Code from Pulse Secure Support and apply it in the Pulse Policy Secure admin console.

Note: From release 9.1R4 onwards, PPS supports VM with 2-NICs model and 3-NICs model for deployment.

Deploying Pulse Policy Secure on Azure Marketplace

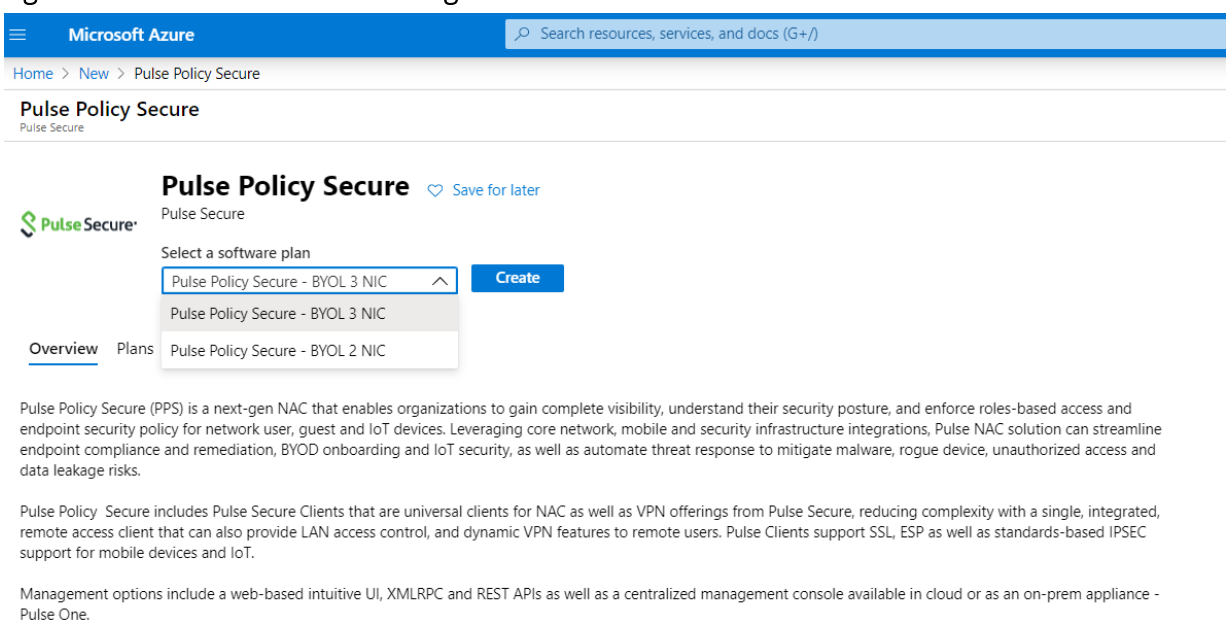
1. Log into Azure portal and navigate to Azure Marketplace by clicking **Create a resource**.

   Figure 1: Marketplace



2. Search with keyword **Pulse Policy Secure**.

   Figure 2: Published Pulse Secure Images



   Azure Marketplace contains the following two Pulse Policy Secure SKUs:

   - Pulse Policy Secure-BYOL 2 NIC
   - Pulse Policy Secure BYOL-3 NIC

3. Select **Pulse Policy Secure BYOL-3 NIC** and click **Create**. In this section, 3-NICs model is chosen as example.

Basic Configuration

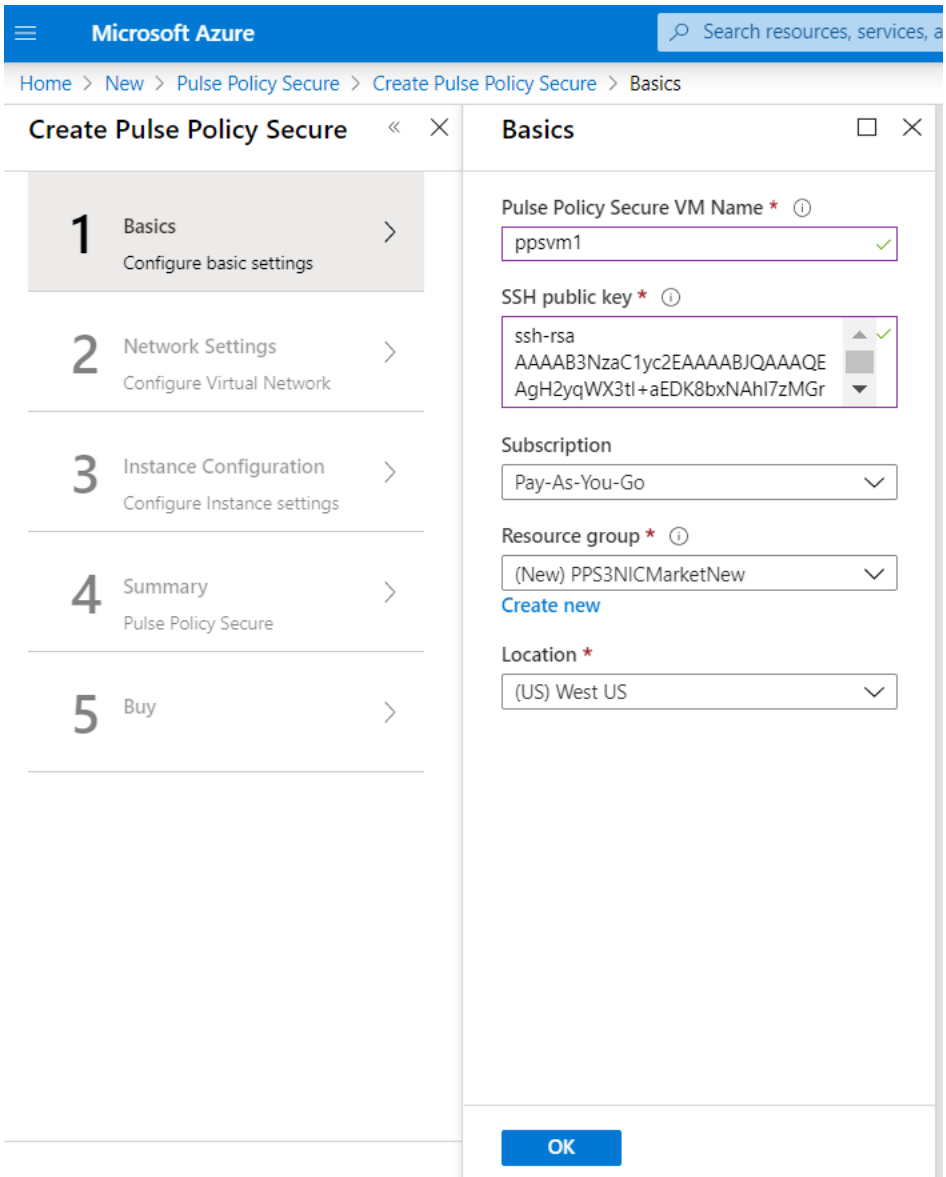
4. In the Basic Configuration step, enter the following parameters and click **OK**:
   - **VM name**: Name of the Pulse Policy Secure to be deployed. Virtual name can be only lower-case letters and numbers, and must be 1-9 characters long.
   - **SSH public key**: Copy and paste an RSA public key in the single-line format or the multi-line PEM format. This key is used to access PPS via SSH. The SSH keys are generated using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer:

   For Windows: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows

For MacOS and Linux: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/mac-create-ssh-keys

- **Resource group**: Create New Resource Group.
- **Location group**: Define the location group.

Figure 3: Basic Configuration Settings



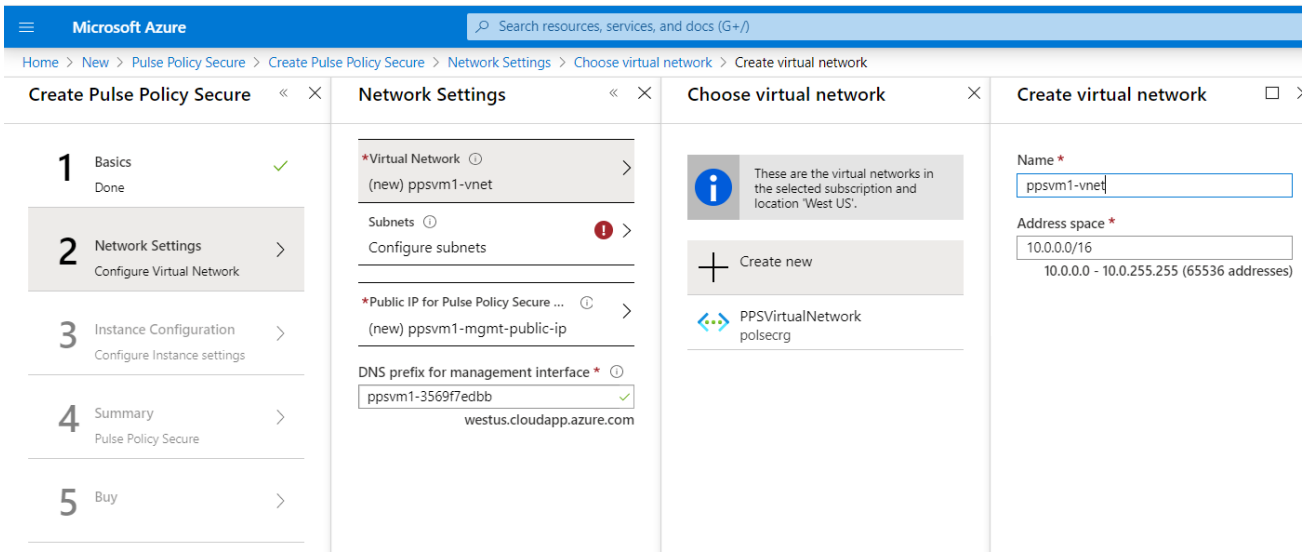
Network Settings

5. In the Network Settings configuration step, enter the following parameters:

   Virtual Network:

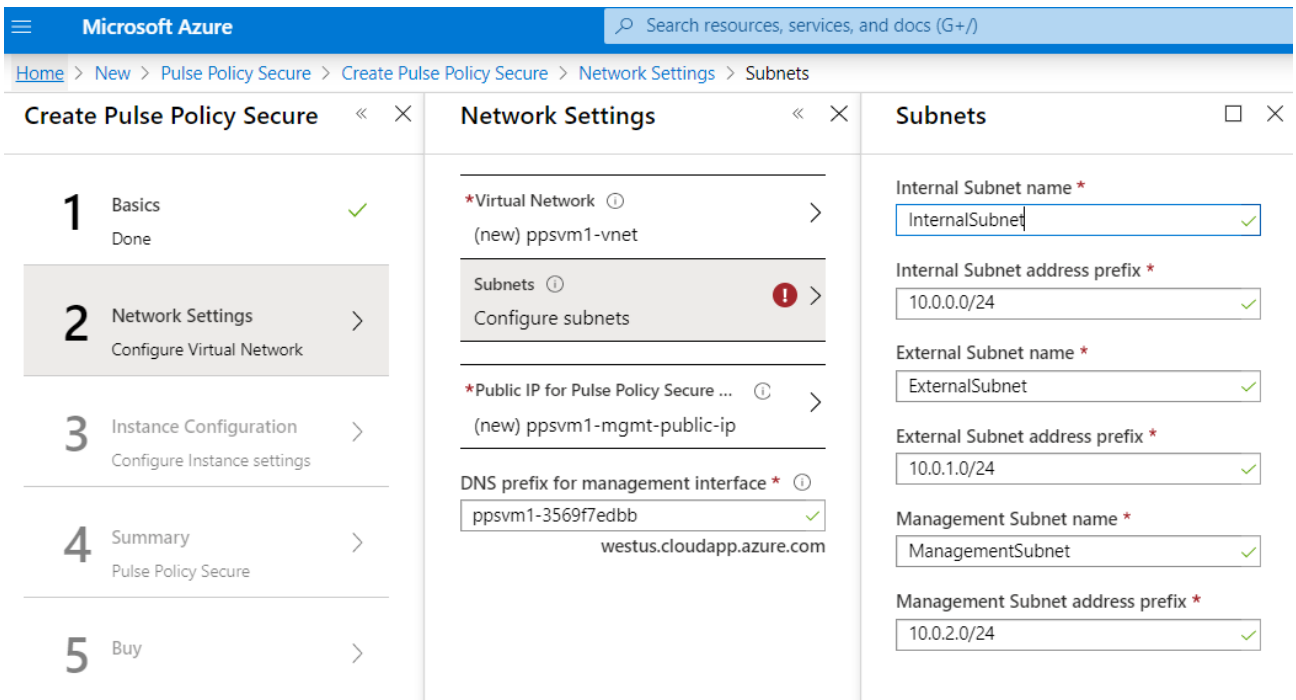
   - Select an existing virtual network from the list or
   - Create a new virtual network. Specify the virtual network name and the address space.

   Figure 4: Virtual Network



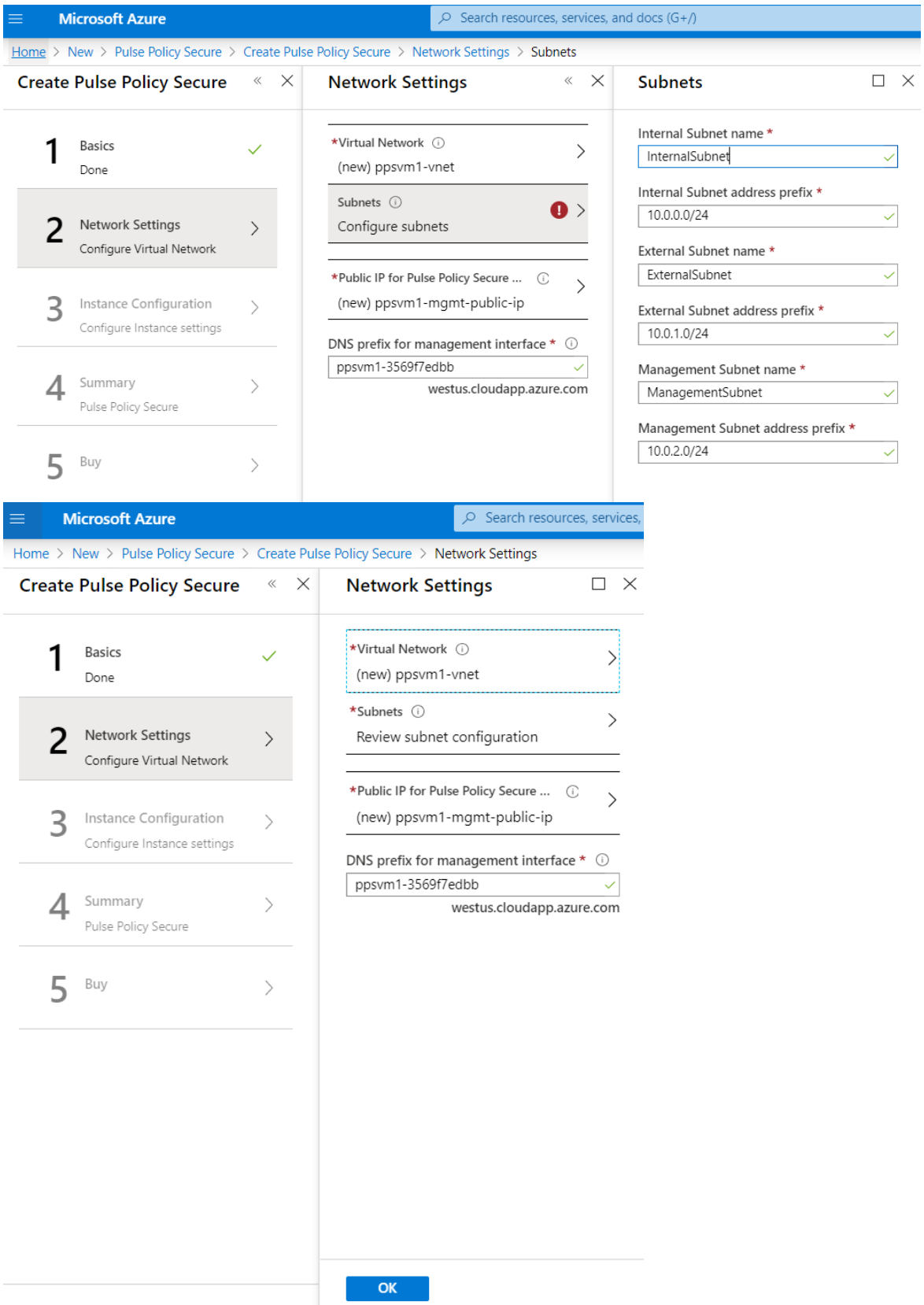
- Subnets: Three subnets – external, internal and management subnets are auto-populated with names and address prefix values. Make any changes if required.

Figure 5: Subnets



- Public IP name and DNS prefix for the Management interface is auto populated. Make any changes if required.
  Note that in a 2-NICs model, Public IP name and DNS prefix name for the Internal interface is auto-populated.

Figure 6: Network Settings

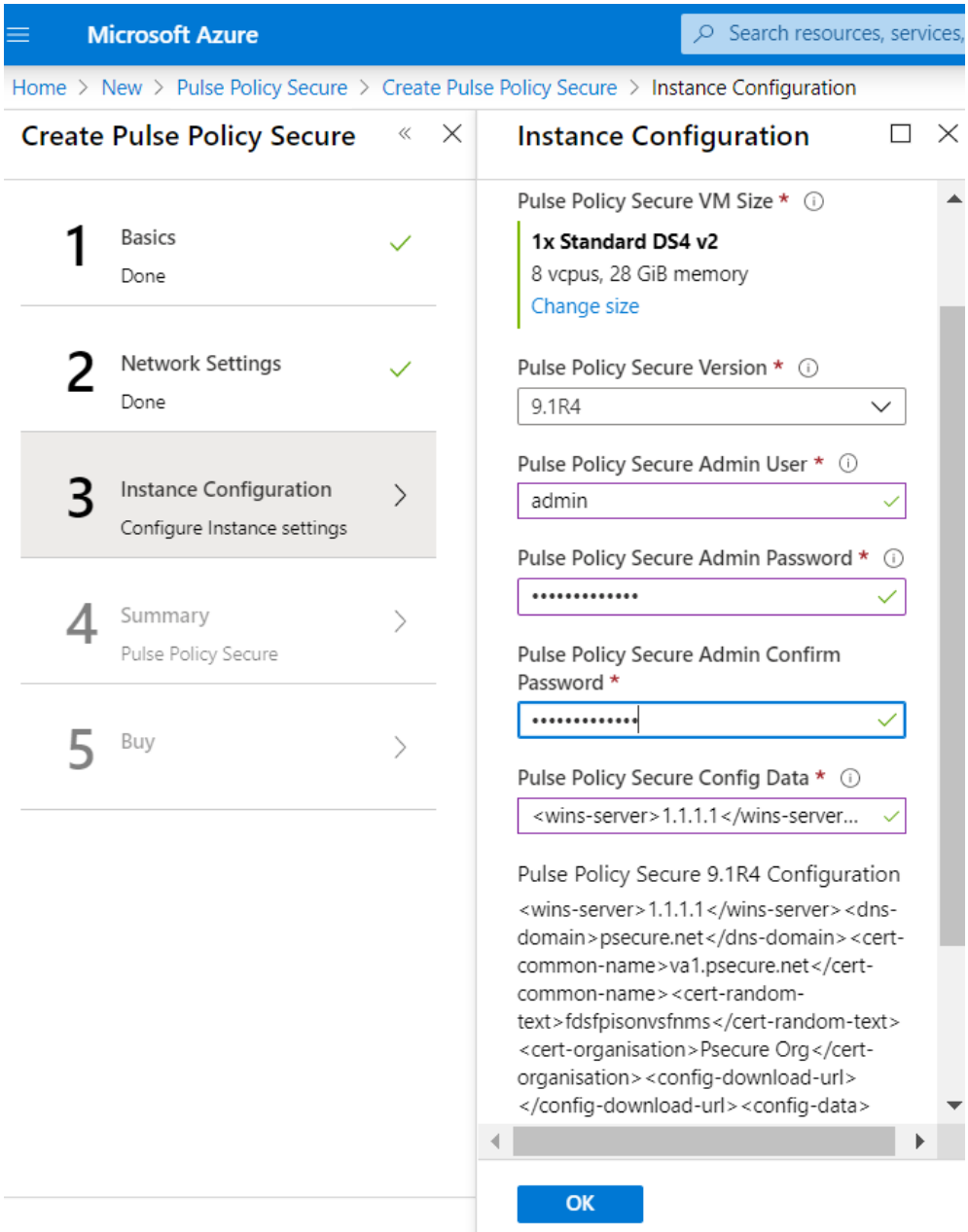


Instance Configuration

6. In the Instance Configuration step, enter the following parameters:
   - **Pulse Policy Secure VM Size**: Specify the size of VM. By default, **1x Standard DS3-v2** is set for 3-NICs model and **1x Standard DS2-v2** is set for 2-NICs model.
   - **Pulse Policy Secure Version**: Specify the Pulse Policy Secure version number.
   - **Pulse Policy Secure Admin User**: Specify the PPS admin user name.
   - **Pulse Policy Secure Admin Password:** Specify the admin password.
   - **Pulse Policy Secure Admin Confirm Password:** Specify the same admin password.
   - **Pulse Policy Secure Config Data:** Provisioning parameters in an XML format. Refer the section "Pulse Policy Secure Provisioning Parameters'

Note: Ensure that the attribute "accept-license-agreement" in pulse-config is set to "y".

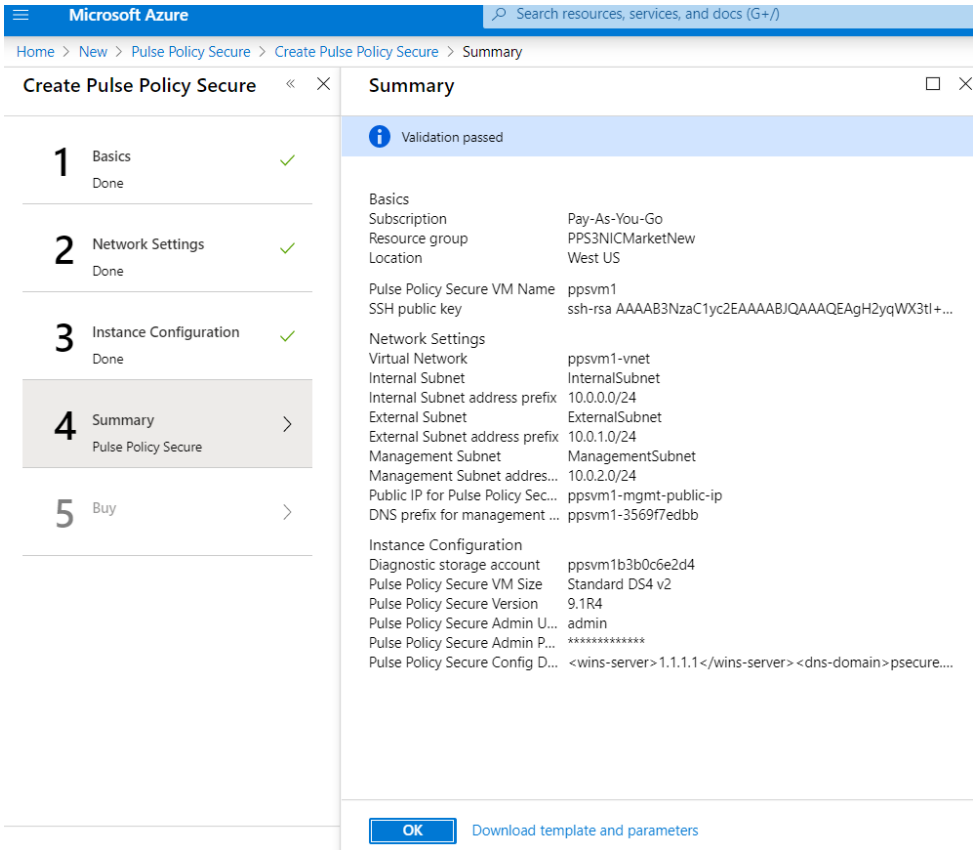
Figure 7: Instance Configuration



Summary Step

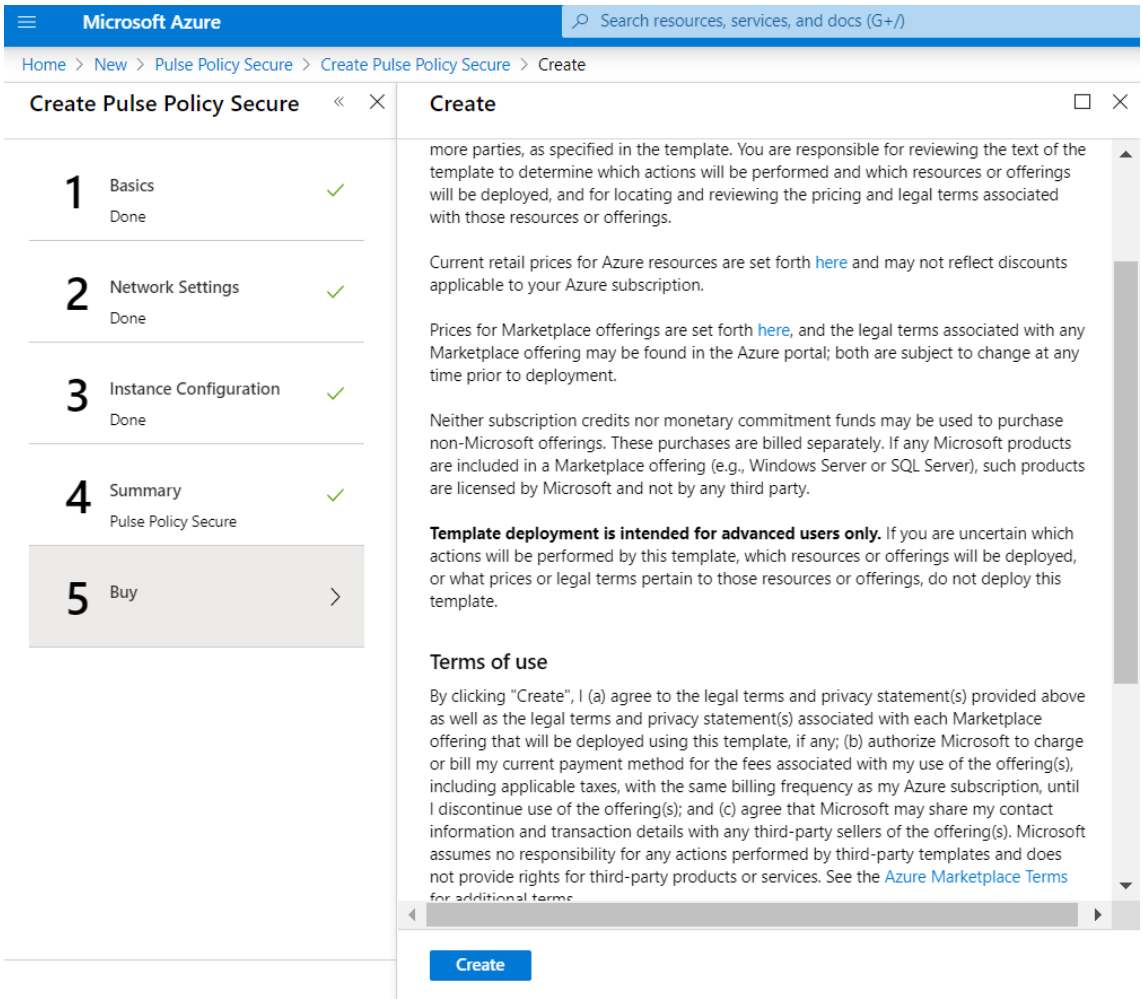
7. In the Summary step, once the final validation is complete, click **OK**.

Figure 8: Configuration Validation



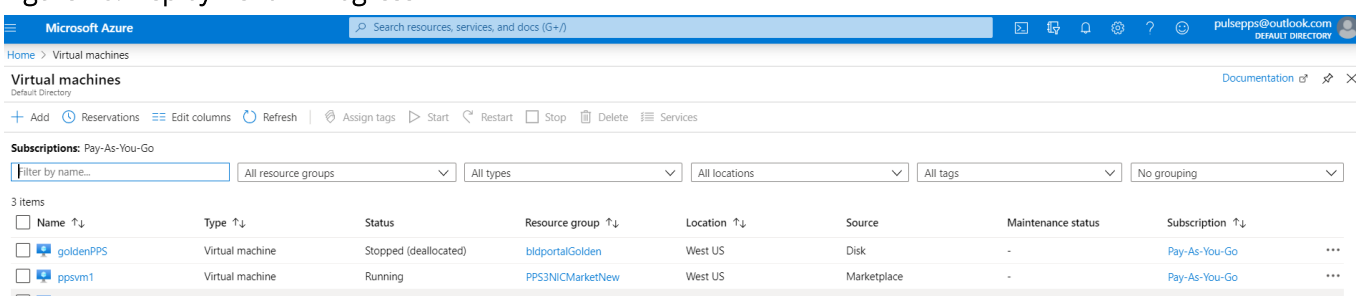
8. Finally, in the Terms of Use page, accept the terms and click **Create**.

Figure 9: Terms of Use



9. Wait for a few minutes while it creates all the resources. This completes deploying PPS on Azure Marketplace.

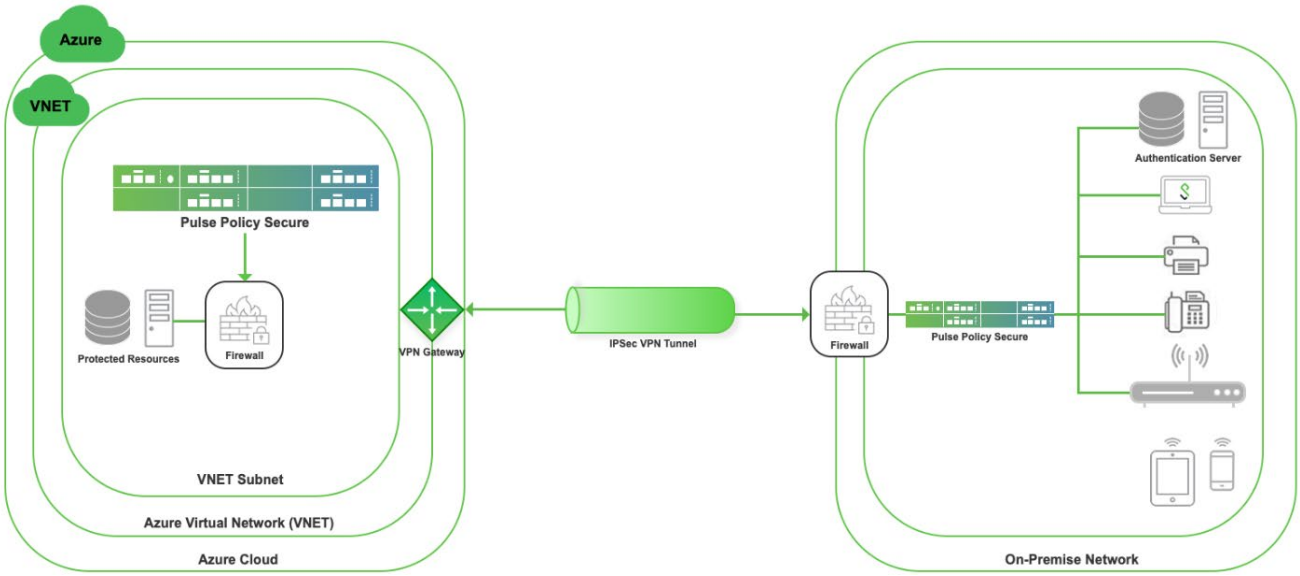
Figure 10: Deployment in Progress



Pulse Policy Secure on Microsoft Azure Cloud

As depicted in the below diagram, an On-premise user can use Pulse Policy Secure to securely access cloud resources as well as corporate resources. To access PPS or cloud resources there should be site-to-site VPN between Azure and corporate network.

Figure 11: Pulse Policy Secure on Microsoft Azure



Prerequisites and System Requirements on Azure

To deploy the Pulse Policy Secure Virtual Appliance on Azure, you need the following:

- A Microsoft Azure account
- Access to the Microsoft Azure portal (https://portal.azure.com)*
- Pulse Policy Secure Virtual Appliance Image (.vhd file)
- Azure Resource Manager template (ARM template)
- Pulse Policy Secure licenses **
- Site-to-Site VPN between Azure and the corporate network since end users are in corporate network.
- Pulse License Server (optional)**
    - Located at corporate network, accessible through site-to-site VPN
- Pulse Policy Secure configuration in XML format (optional)
- The following systems are qualified in 9.1R4 release:
    - DS2 – 2-core
    - DS3 – 4-core
    - DS4 – 8-core

Note:

* Pulse Policy Secure Virtual Appliance can be deployed only through Azure Resource Manager (ARM) style. It does not support deployment in classic style.

** Pulse Policy Secure Virtual Appliance, by default, has two-users license. This release supports licensing with License server located at corporate network and licensing through Pulse Cloud Licensing Service (PCLS) server. For licensing through PCLS, administrator needs to obtain Authentication Code from Pulse Secure Support and apply it in the Pulse Policy Secure admin

console.

Steps to Deploy Pulse Policy Secure on Azure

Below are the one-time activities to be followed to deploy Pulse Policy Secure on Azure.

- Upload Pulse Policy Secure Virtual Appliance Image to Azure Web Portal
- Upload Azure Resource Manager Template to Azure Account

Below are the steps to be followed for each deployment of Pulse Policy Secure.

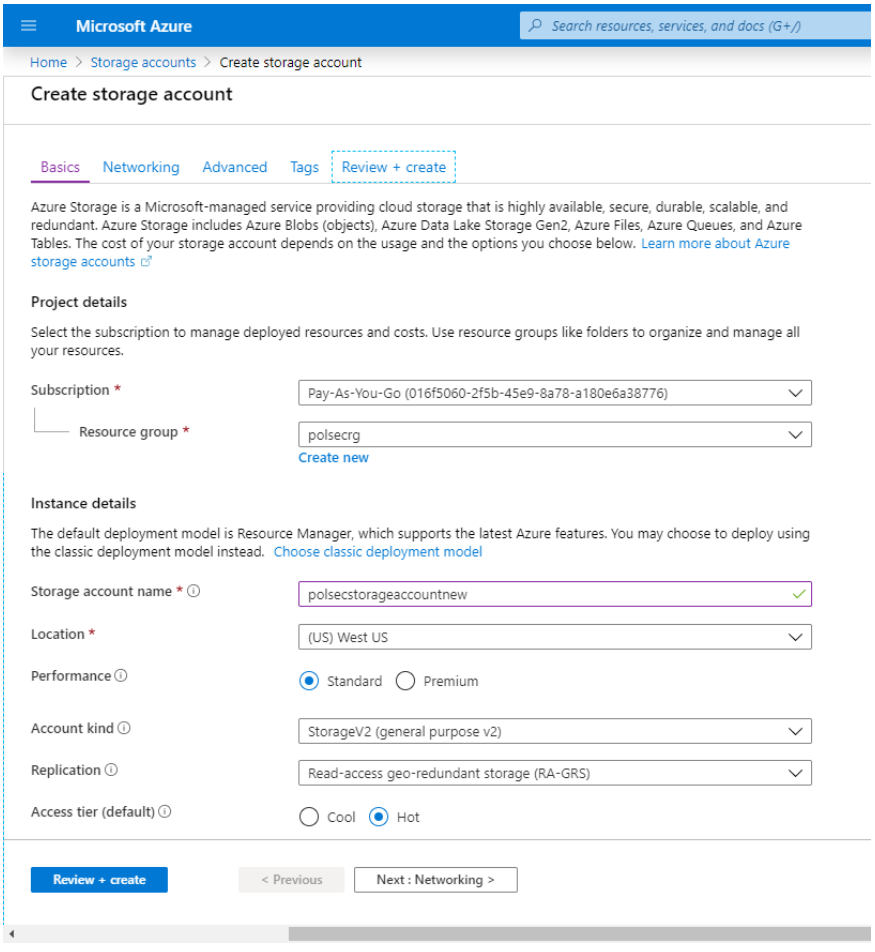
- Deploying Pulse Policy Secure on Azure using Azure Portal
- Deploying Pulse Policy Secure on Azure using Azure CLI

Upload Pulse Policy Secure Virtual Appliance Image to Azure Web Portal

This section shows the steps to upload the Pulse Policy Secure Virtual Appliance image to Azure web portal.
To upload Pulse Policy Secure Virtual Appliance image to Azure web portal, do the following:

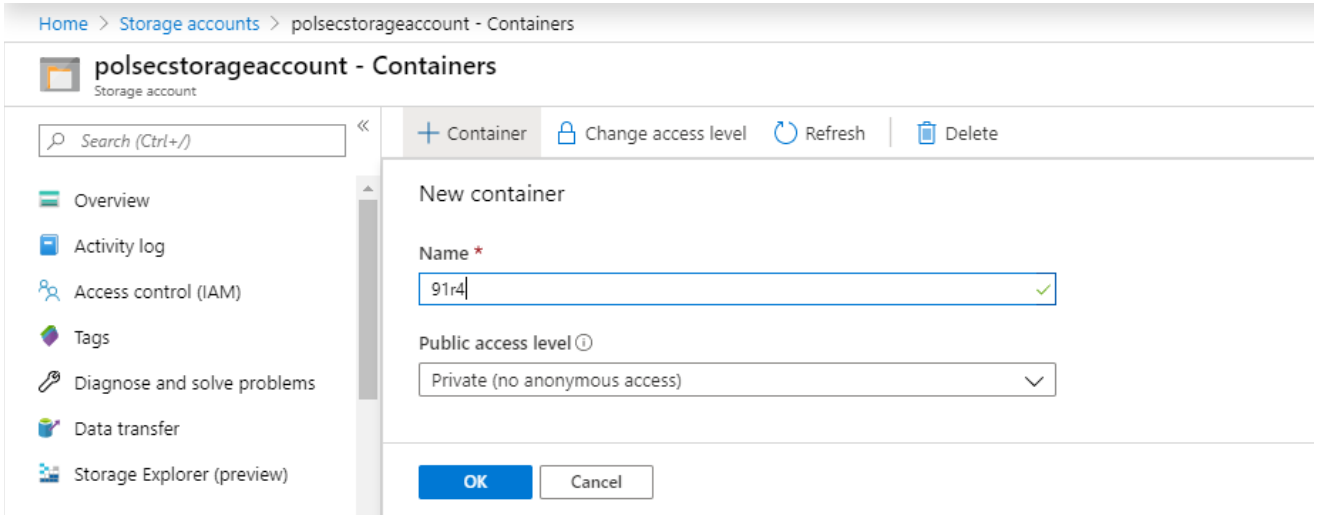
1. Visit the Pulse Secure support site www.pulsesecure.net and download the Azure PPS image file (*ps-pps-azure-psa-v-<releaseno>-<buildno>-package.zip*) which is in the zipped format.
2. Unzip the file and look for the Pulse Policy Secure Virtual Appliance **vhd** image.
3. Log in to the Azure portal.
4. Click **New** and create a storage account named 'polsecstorageaccountnew' under the resource group named 'polsecrg'.

Figure 12: Storage Account - polsecstorageaccountnew



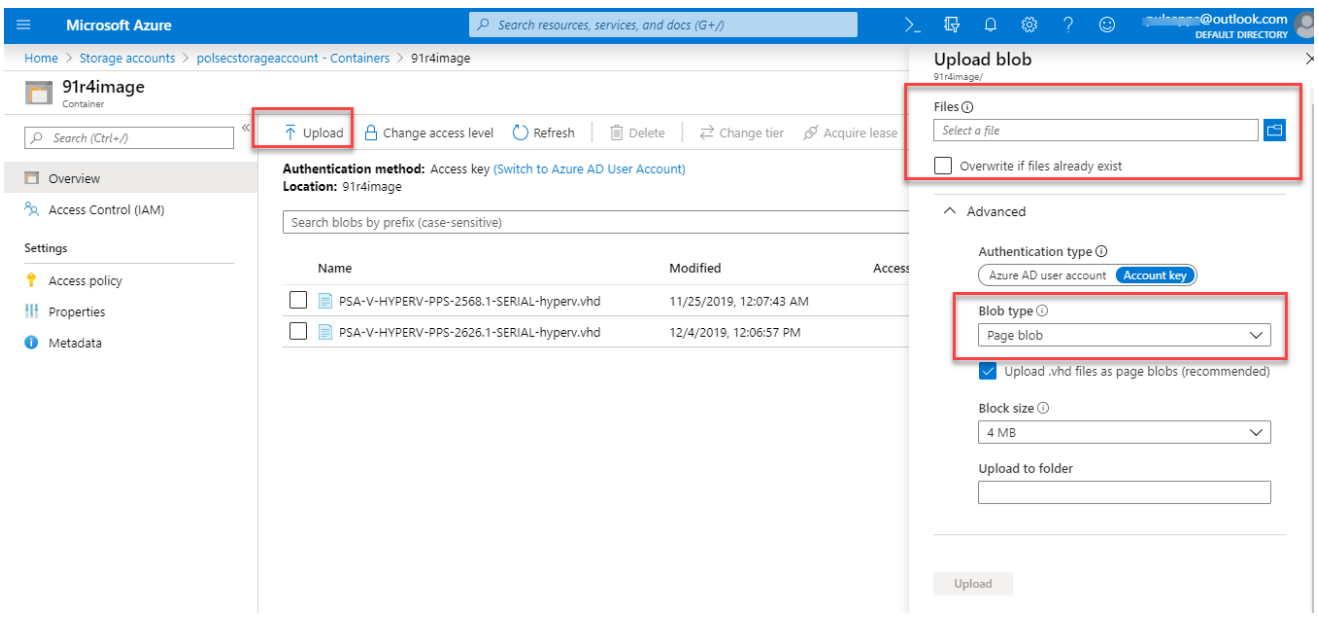
5. Inside the 'polsecstorageaccountnew' storage account, create a container named '91r4' as shown below -

Figure 13: Container



6. Inside the '91r4' container, click on **upload** to upload the Pulse Policy Secure Virtual Appliance image. Inside the 'Upload blob', select the Blob type as Page blob and click on **Upload**.

Figure 14: Upload Pulse Policy Secure Virtual Appliance Image



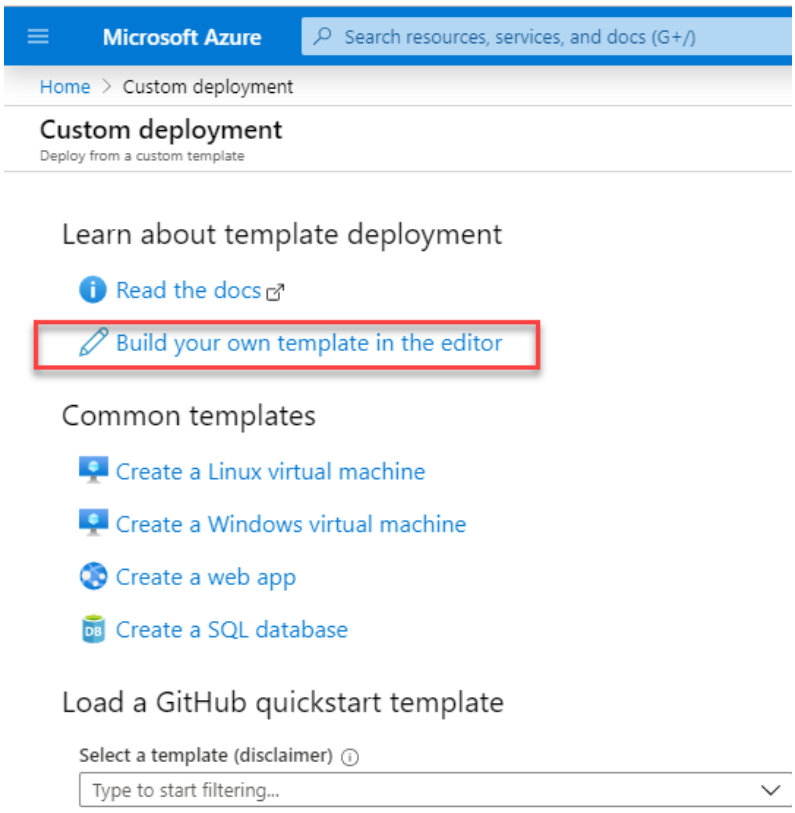
Upload Azure Resource Manager Template to Azure Account

The Azure Resource Manager (ARM) template is a JSON-based file, which has instructions for Azure Fabric on all the resources that need to be created on Azure while running this script. More details on the ARM template can be found at https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-create-first-template.

Pulse Secure provides sample Azure template file for three NIC card, namely "pulsesecure-pps-3-nics.zip" and "pulsesecure-pps-3-nics-existing-vnet.zip". Users can modify the template to make it suitable for their need. Here are the steps to upload the template to Azure Portal.

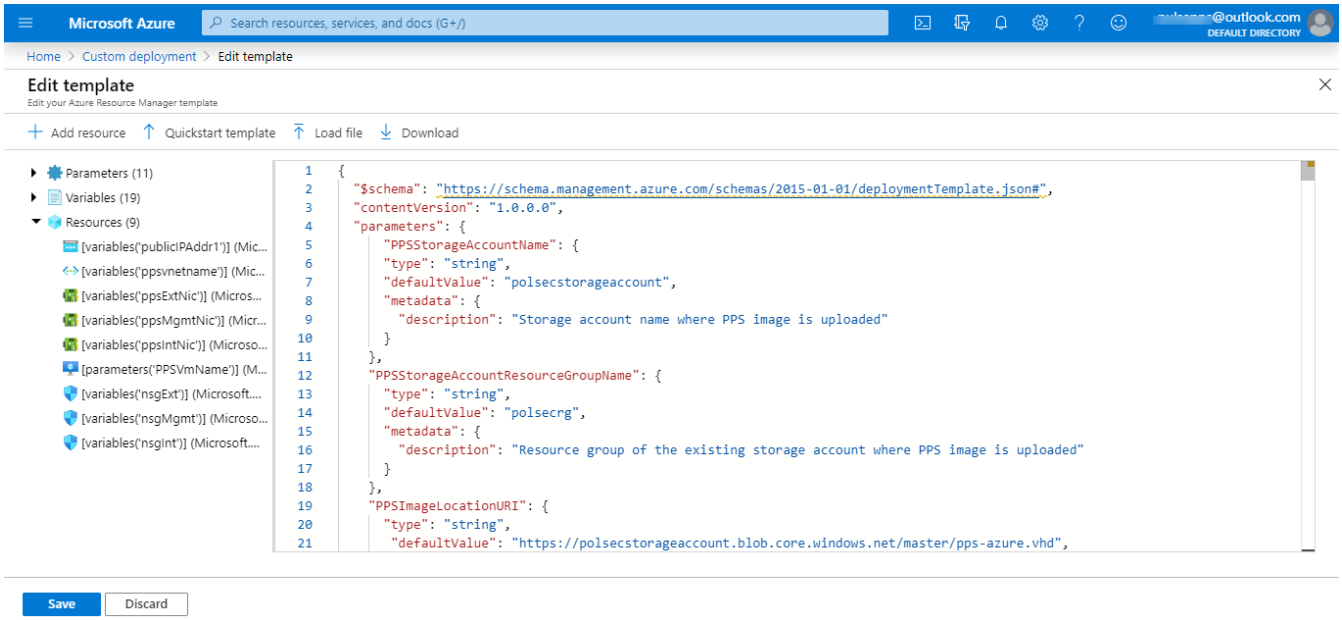
1. Unzip the **pulsesecure-pps-3-nics.zip** file to get **azuredeploy.json.**
2. Log in to the Azure portal.
3. In the search bar type 'Deploy from a custom template'. Click on 'Build your own template in the editor'.

Figure 15: Deploy from a custom template



4. Copy the contents of **azuredeploy.json** and paste it in the template section. Click on 'Save' Button.

Figure 16: Add ARM Template



Deploying Pulse Policy Secure on Azure using Azure Portal

Before proceeding with the deployment, refer the following sections:

- Upload Pulse Policy Secure Virtual Appliance Image to Azure
- Upload Azure Resource Manager Template to Azure Account

Pulse Policy Secure can be deployed on:

- a new Virtual network or
- an already existing Virtual network

Deploying PPS on New Virtual Network

This section describes deployment with three NIC cards and two NIC cards.

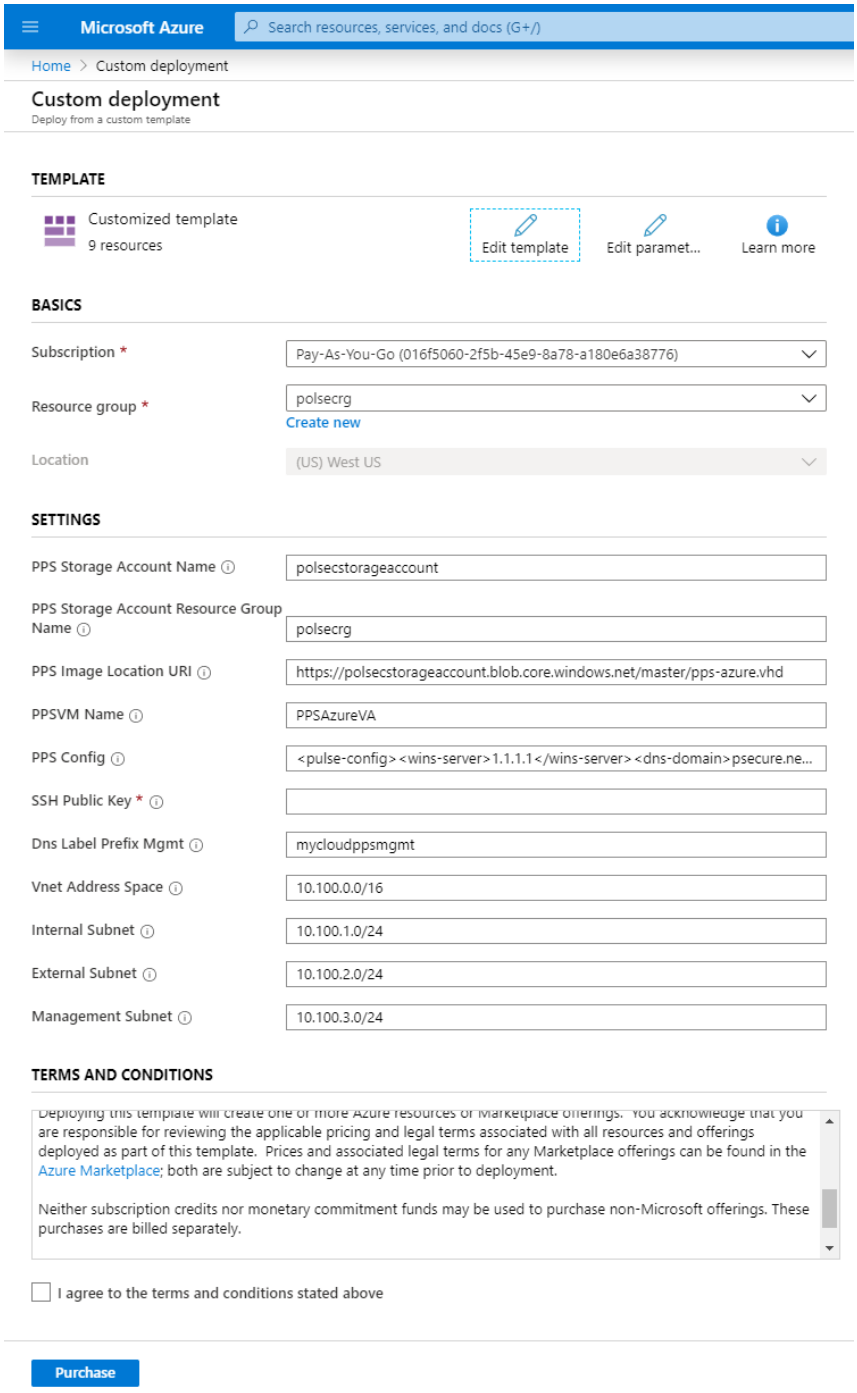
Deployment on VM with Three NIC Cards

To deploy PPS on Azure using the Azure portal, do the following:

1. Select the template file created in section 'Upload Azure Resource Manager Template to Azure account' and click **Save**.

Note: Before proceeding with deployment, ensure that the attribute "**accept-license-agreement"** in pulse-config is set to "**y**".

Figure 17: Custom Deployment on VM with Three NIC Cards – New Virtual Network



2. Fill or modify the following parameters:
   - **Resource group:** Specify the resource group name in which Pulse Policy Secure needs to be deployed
   - **Location:** Region where resource group needs to be created
   - **PPS Storage Account Name:** Storage account name where the Pulse Policy Secure Virtual Appliance image is available
   - **PPS Storage Account Resource Group:** Resource group of where the Pulse Policy Secure Virtual Appliance image is copied
   - **PPS Image Location URI:** URI to Pulse Policy Secure Virtual Appliance Image
   - **PPSVM Name:** Name of the Pulse Policy Secure Virtual instance

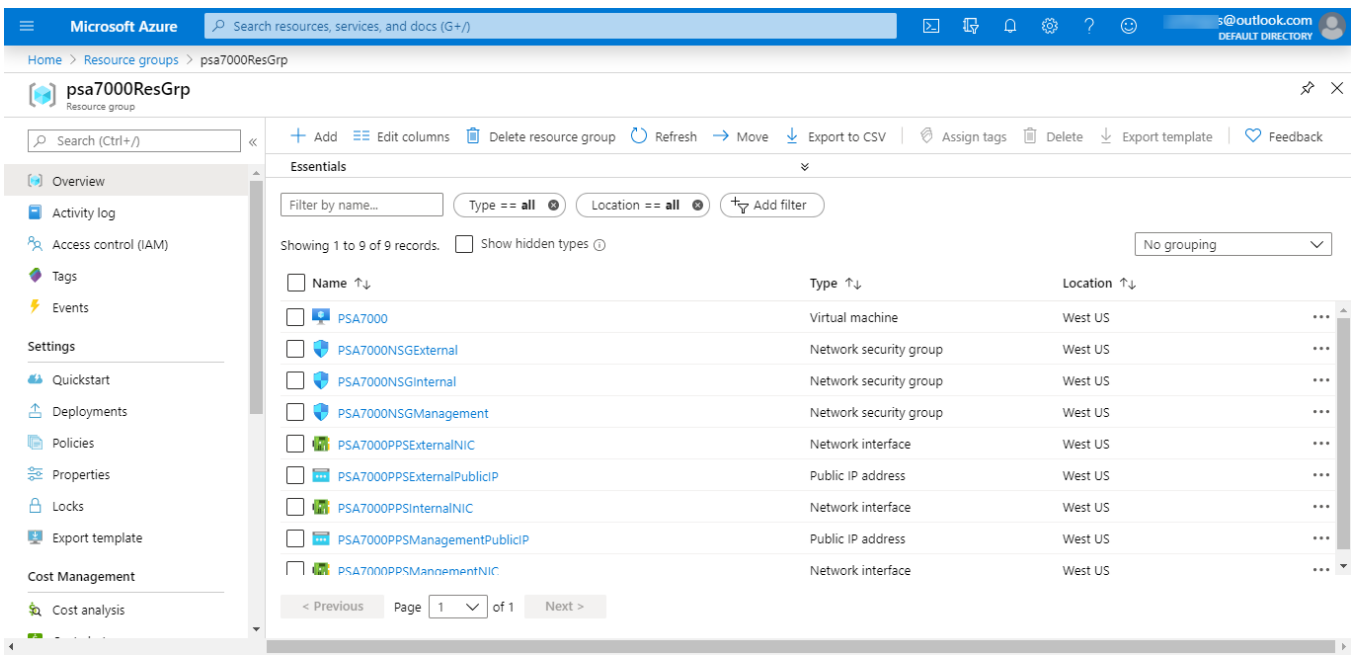
- **PPS Config:** Provisioning parameters in an XML format. Refer the section '[Pulse Policy Secure Provisioning Parameters]'
- **SSH Public Key:** This key is used to access PPS via SSH. The SSH keys are generated using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer:

For Windows: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows
For MacOS and Linux: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/mac-create-ssh-keys

- **DNS Label Prefix Mgmt:** Prefix for the management interface DNS label
- **Vnet Address Space:** Virtual network address space
- **Internal Subnet:** Subnet from which Pulse Policy Secure internal interface needs to lease IP
- **External Subnet:** Subnet from which Pulse Policy Secure external interface needs to lease IP
- **Management Subnet:** Subnet from which Pulse Policy Secure management interface needs to lease IP

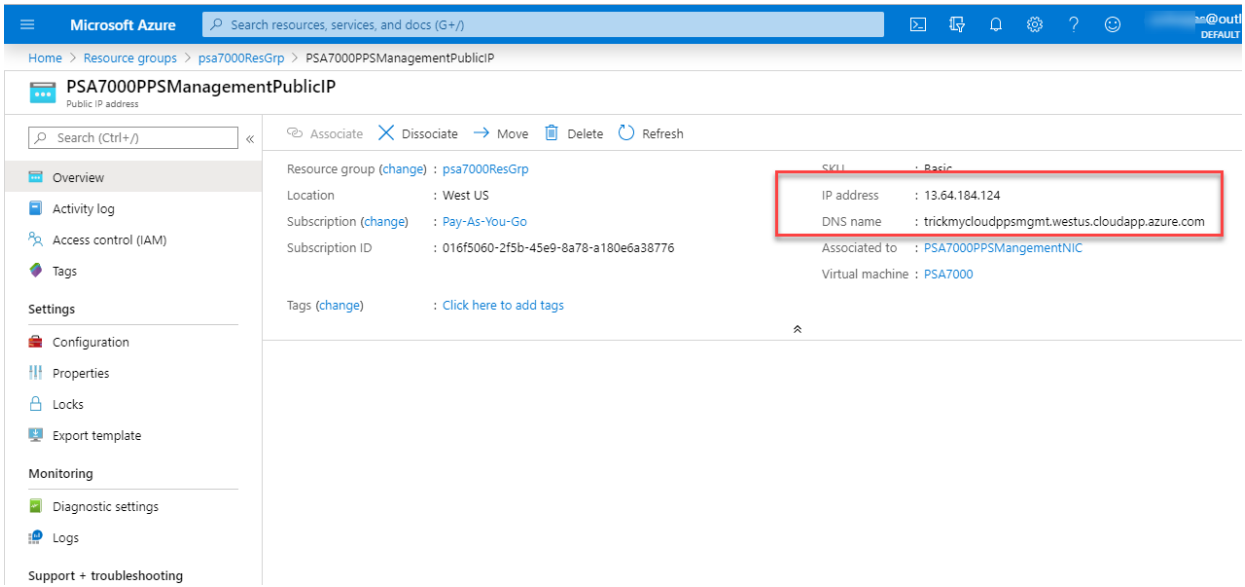
3. Agree to the Azure licensing terms and click **Purchase**.
4. Watch for the deployment succeeded message after 3 to 5 minutes.

Figure 18: Deployment Succeeded



5. Go to the resource group in which the Pulse Policy Secure Virtual Appliance was deployed to see the resources created.
6. Navigate to the resource group and click **PPS Management Public IP**. Make a note of the PPS Management Public IP and DNS name (FQDN) to access PPS for admin page.

Figure 19: PPS Management Public IP



Note: Azure allows static as well as dynamic assignment of IP addresses to the network interfaces. The mode of IP assignment (static/dynamic) can be mentioned in the Azure Resource Manage template file. The current JSON template uses the dynamic method of allotting IP addresses to the network interfaces.

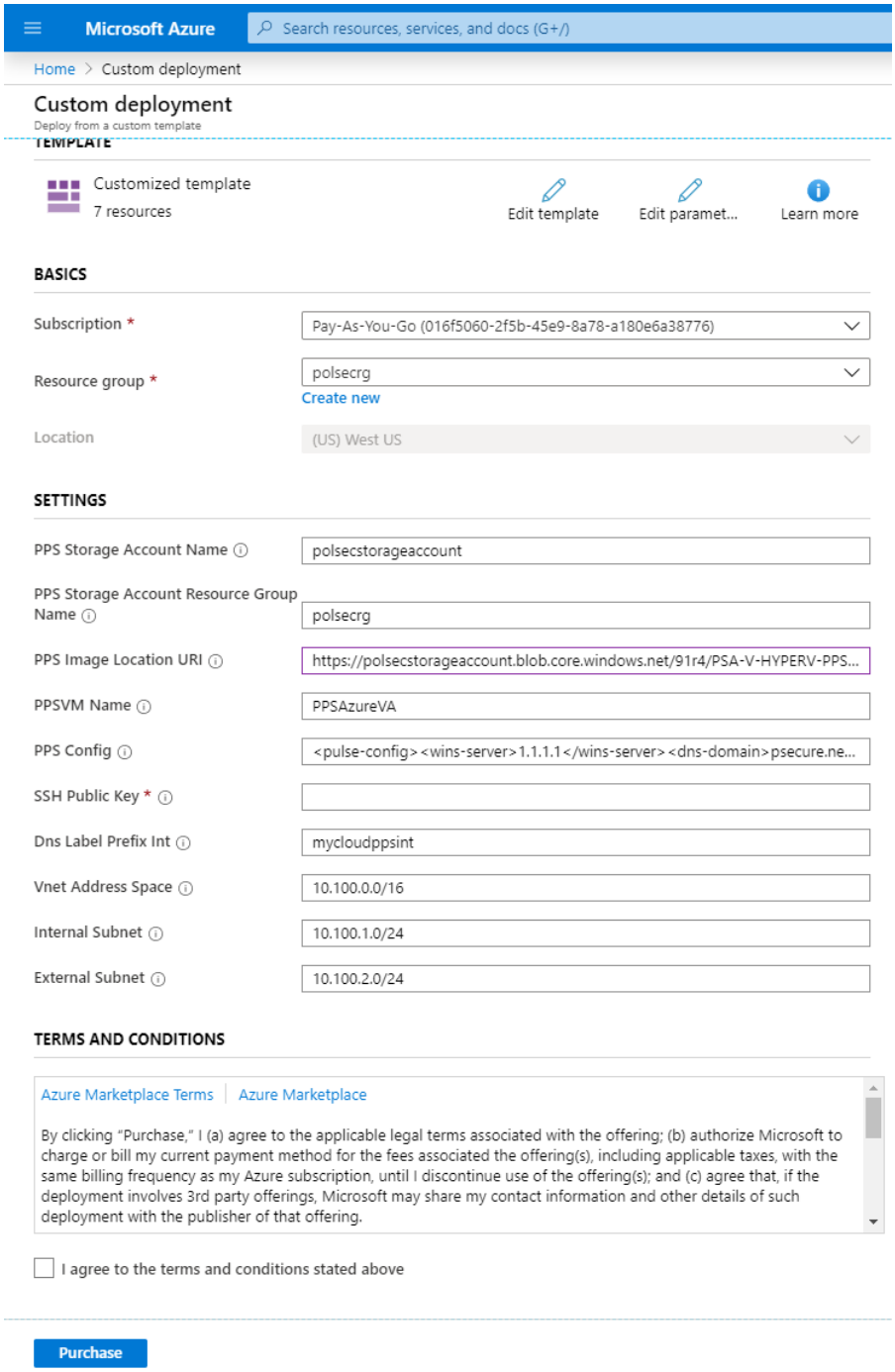
Deployment on VM with Two NIC Cards

To deploy Pulse Policy Secure on Azure using the Azure portal, do the following:

1. Select the template file created in section 'Upload Azure Resource Manager Template to Azure account' and click **Deploy**.

Note: Before proceeding with deployment, ensure that the attribute "**accept-license-agreement"** in pulse-config is set to "**y**".

Figure 20: Custom Deployment on VM with Two NIC Cards – New Virtual Network



2. Fill or modify the following parameters:
   - **Resource group:** Specify the resource group name in which Pulse Policy Secure needs to be deployed
   - **Location:** Region where resource group needs to be created
   - **PPS Storage Account Name:** Storage account name where the Pulse Policy Secure Virtual Appliance image is available
   - **PPS Storage Account Resource Group:** Resource group of where the Pulse Policy Secure Virtual Appliance image is copied
   - **PPS Image Location URI:** URI to Pulse Policy Secure Virtual Appliance Image
   - **PPSVM Name:** Name of the Pulse Policy Secure Virtual instance
   - **PPS Config:** Provisioning parameters in an XML format.

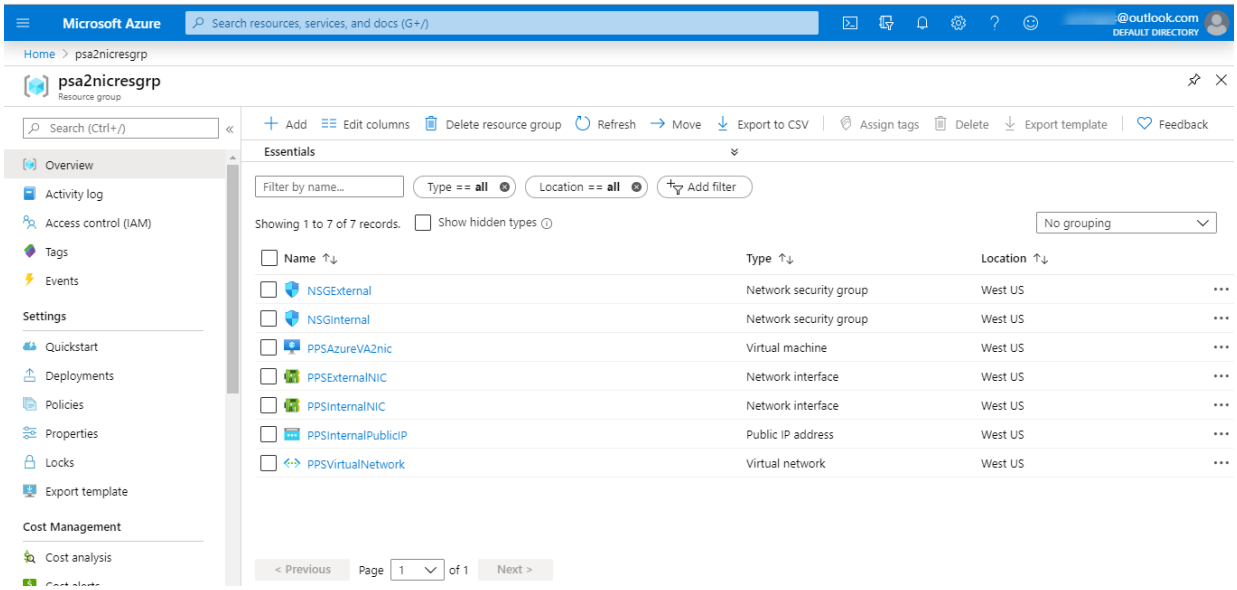
- **SSH Public Key:** This key is used to access PPS via SSH. The SSH keys are generated using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer:

For Windows: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows
For MacOS and Linux: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/mac-create-ssh-keys

- **DNS Label Prefix Int:** Prefix for the internal interface DNS label
- **Vnet Address Space:** Virtual network address space
- **Internal Subnet:** Subnet from which Pulse Policy Secure internal interface needs to lease IP
- **External Subnet:** Subnet from which Pulse Policy Secure external interface needs to lease IP

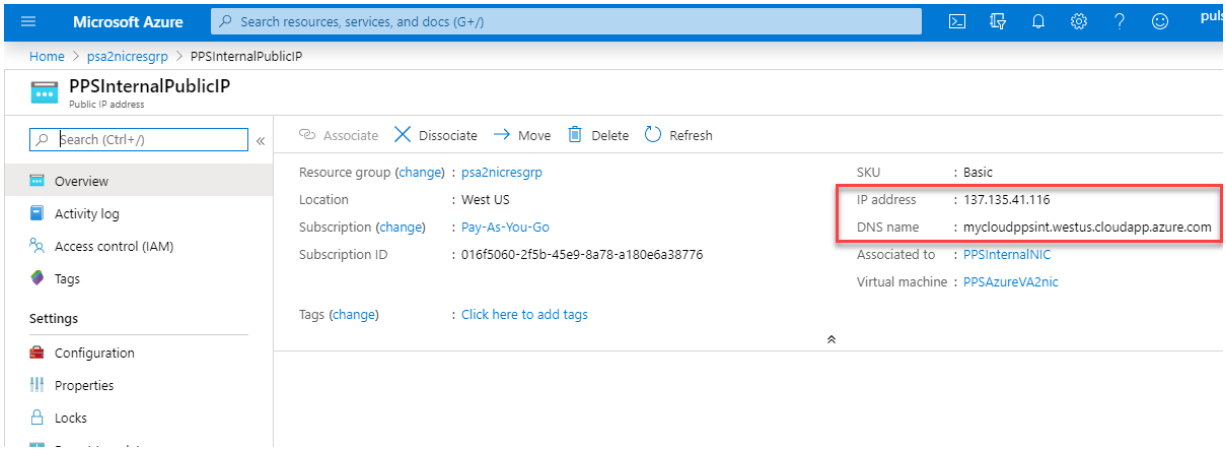
3. Agree to the Azure licensing terms and click **Purchase**.
4. Watch for the deployment succeeded message after 3 to 5 minutes.

Figure 21: Deployment Succeeded



5. Go to the resource group in which the Pulse Policy Secure Virtual Appliance was deployed to see the resources created.
6. Click **PPS Internal Public IP** and note down the **PPS Internal Public IP and DNS name (FQDN)** to access PPS for end user page.

Figure 22: PPS Imternal Public IP



Note: Azure allows static as well as dynamic assignment of IP addresses to the network interfaces. The mode of IP assignment (static/dynamic) can be mentioned in the Azure Resource Manage template file. The current JSON template uses the dynamic method of allotting IP addresses to the network interfaces.

Deploying PPS on an Existing Virtual Network

This section describes deployment with three NIC cards and two NIC cards.

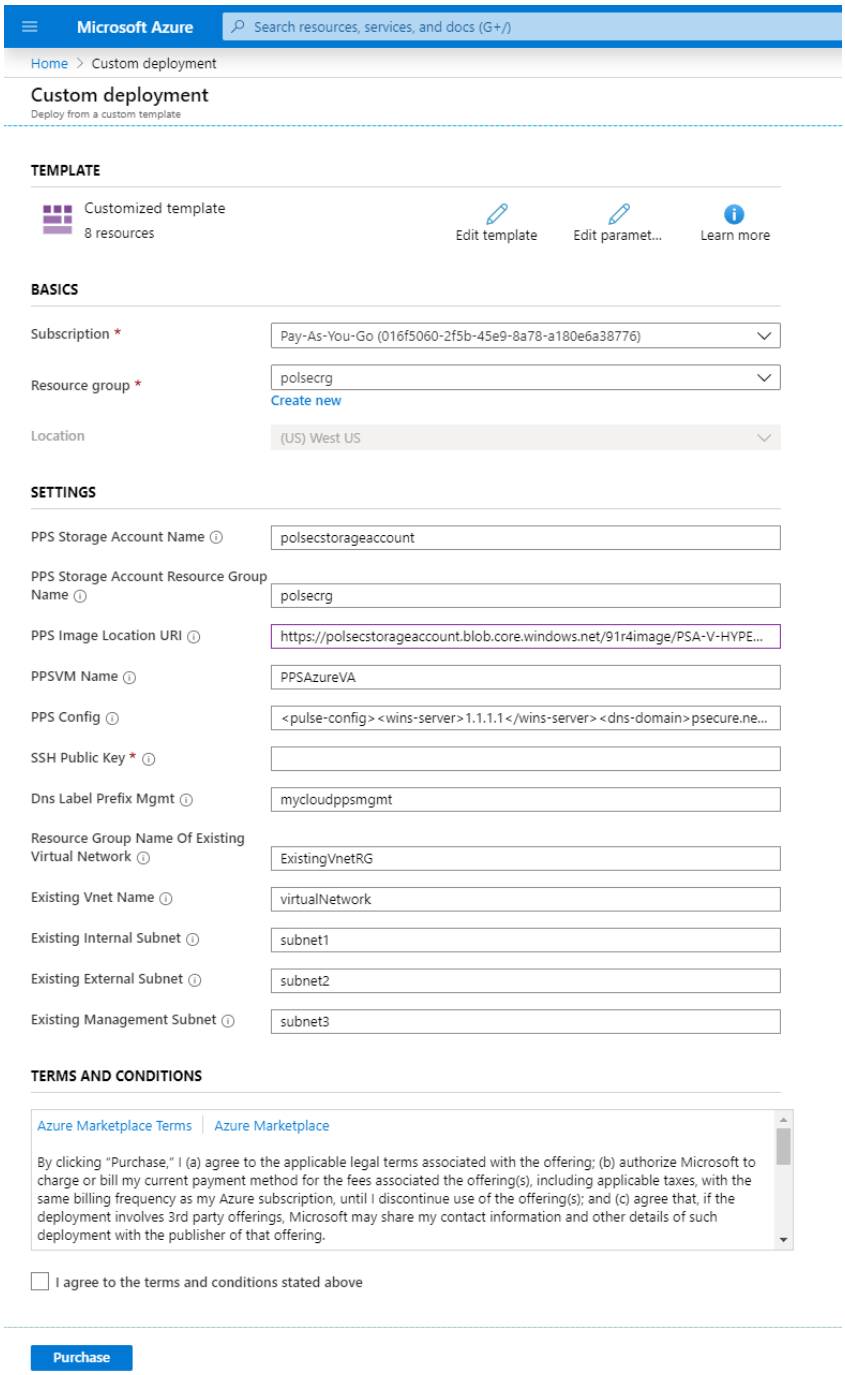
Deployment on VM with Three NIC Cards

To deploy Pulse Policy Secure on Azure using the Azure portal, do the following:

1. Select the template file "pulsesecure-pps-3-nics-existing-vnet" created in the section 'Upload Azure Resource Manager Template to Azure account' and click **Save**.

Note: Before proceeding with deployment, ensure that the attribute "**accept-license-agreement"** in pulse-config is set to "**y**".

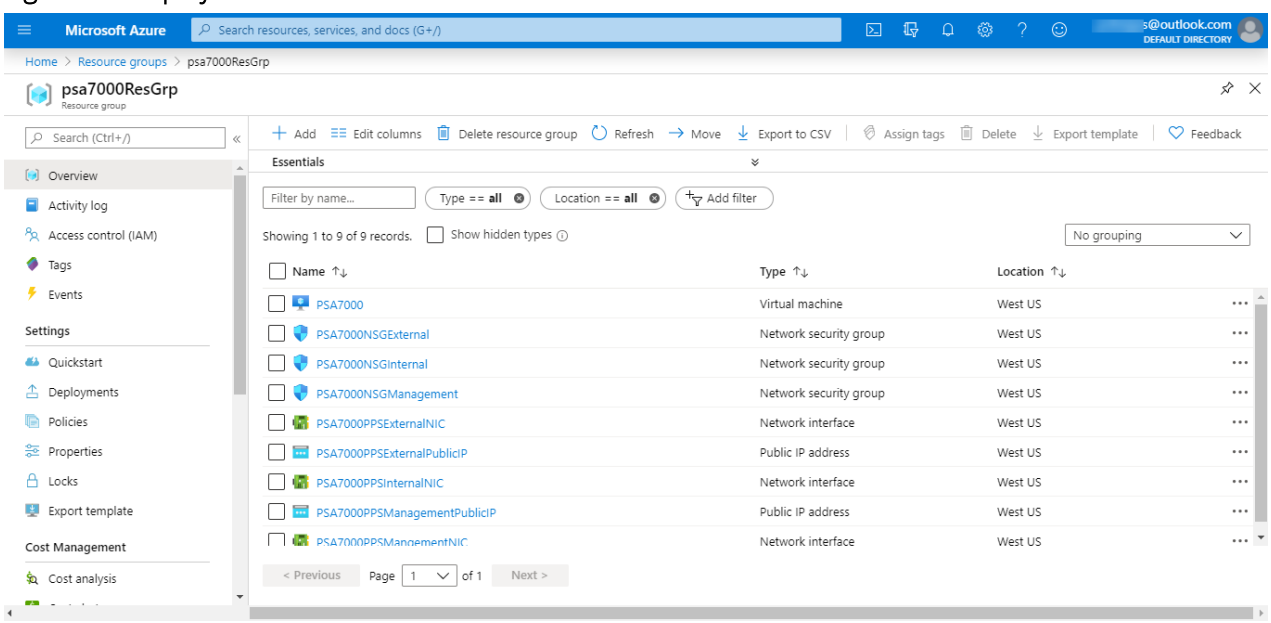
Figure 23: Custom Deployment on VM with Three NIC Cards – Existing Virtual Network



2. Fill or modify the following parameters:
   - **Resource group:** Specify the resource group name in which Pulse Policy Secure needs to be deployed
   - **Location:** Region where resource group needs to be created
   - **PPS Storage Account Name:** Storage account name where the Pulse Policy Secure Virtual Appliance image is available
   - **PPS Storage Account Resource Group:** Resource group of where the Pulse Policy Secure Virtual Appliance image is copied
   - **PPS Image Location URI:** URI to Pulse Policy Secure Virtual Appliance Image
   - **PPS VM Name:** Name of the Pulse Policy Secure Virtual instance
   - **PPS Config:** Provisioning parameters in XML format. Refer 'Pulse Policy Secure Provisioning Parameters'
   - **SSH Public Key:** This key is used to access PPS via SSH. The SSH keys are generated using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer:

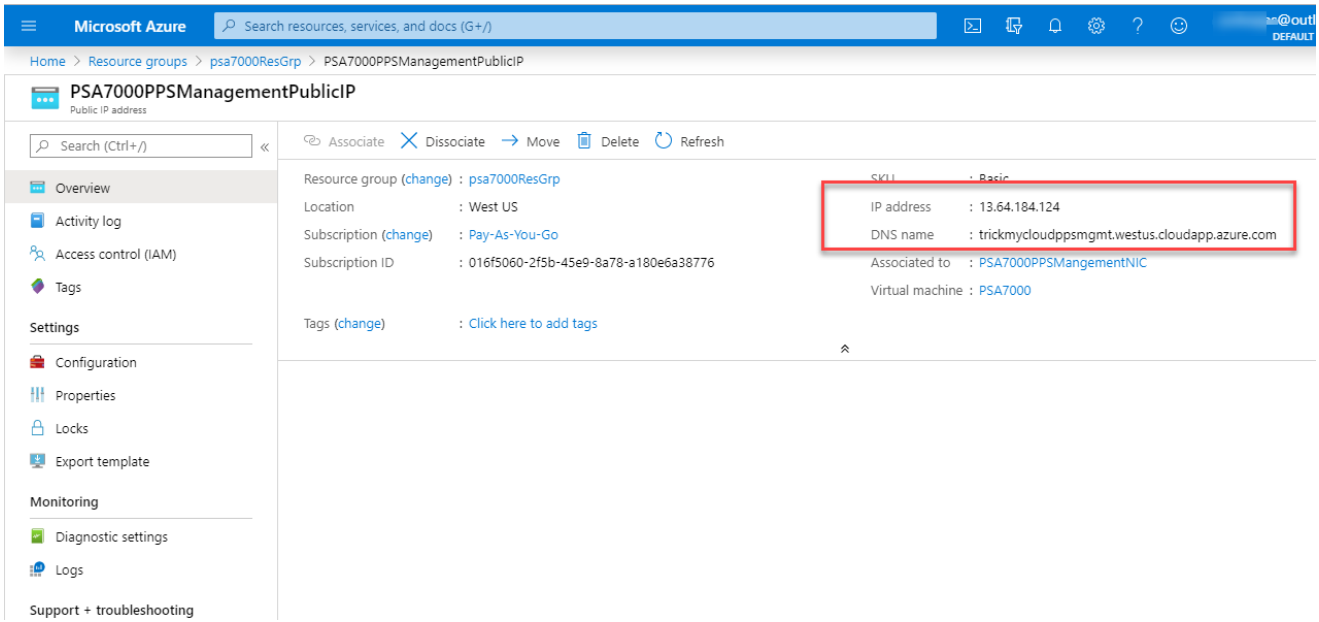
   For Windows: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows
   For MacOS and Linux: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/mac-create-ssh-keys
   - **DNS Label Prefix Mgmt:** Prefix for the management interface DNS label
   - **Resource Group Name of Existing Virtual Network:** Resource Group name of the Virtual network
   - **Existing Vnet Name:** Virtual network name
   - **Existing Internal Subnet:** Subnet from which the Pulse Policy Secure internal interface needs to lease IP
   - **Existing External Subnet:** Subnet from which the Pulse Policy Secure external interface needs to lease IP
   - **Existing Management Subnet:** Subnet from which the Pulse Policy Secure management interface needs to lease IP
3. Agree to the Azure licensing terms and click **Purchase**.
4. Watch for the deployment succeeded message after 3 to 5 minutes.

Figure 24: Deployment Succeeded



5. Go to the resource group in which the Pulse Policy Secure Virtual appliance was deployed to see the resources created.
6. Navigate to the resource group and click **PPS Management Public IP**. Make a note of the PPS Management Public IP and DNS name (FQDN) to access PPS for admin page.

Figure 25: PPS Management Public IP



Note: Azure allows static as well as dynamic assignment of the IP addresses to the network interfaces. The mode of IP assignment (static/dynamic) can be mentioned in the Azure Resource Manage template file. The current JSON template uses dynamic method of allotting IP addresses to the network interfaces.

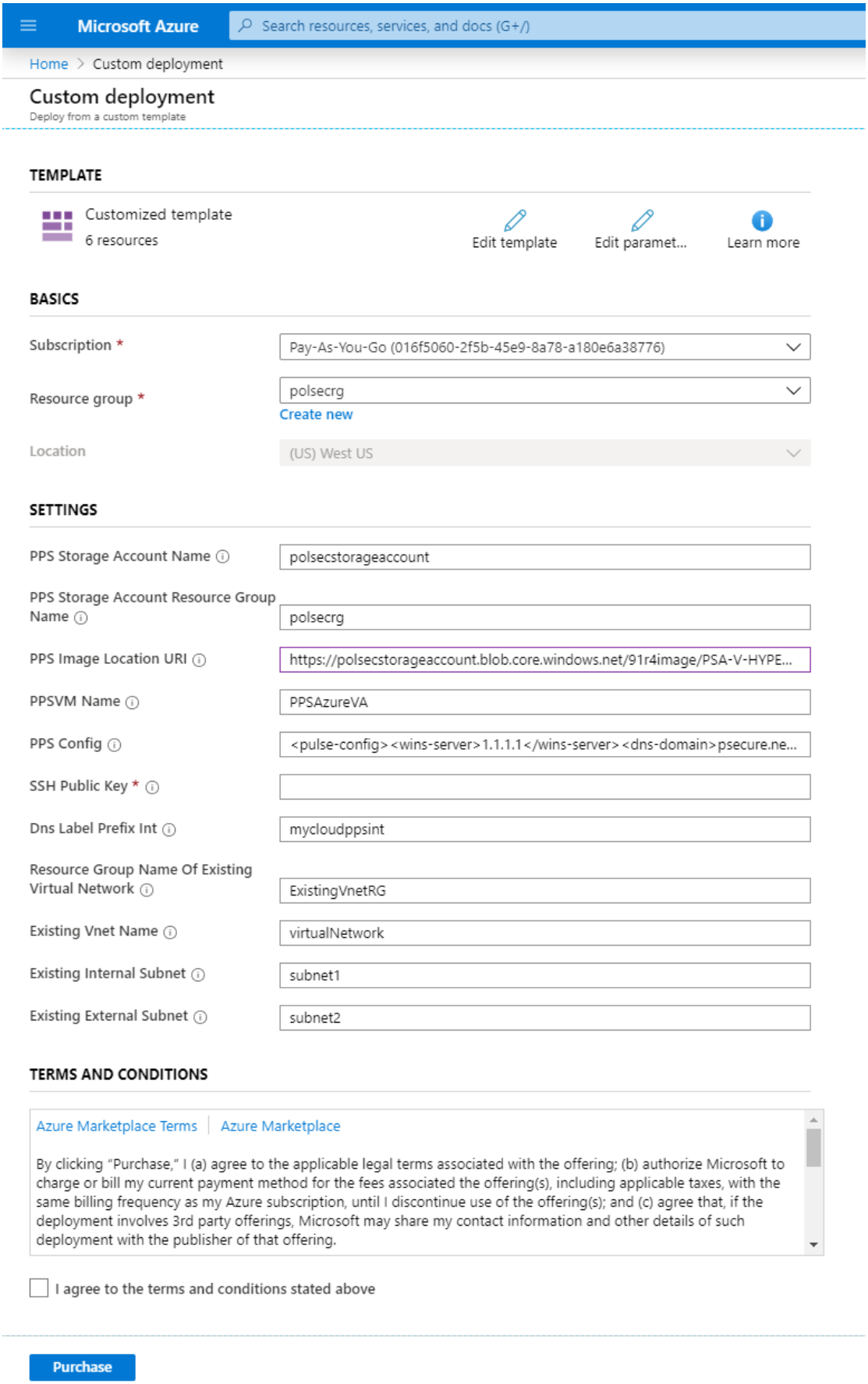
Deployment on VM with Two NIC Cards

To deploy Pulse Policy Secure on Azure using the Azure portal, do the following:

7. Select the template file "pulsesecure-pps-2-nics-existing-vnet" created in the section 'Upload Azure Resource Manager Template to Azure account' and click **Deploy**.

Note: Before proceeding with deployment, ensure that the attribute "**accept-license-agreement"** in pulse-config is set to "**y**".

Figure 26: Custom Deployment on VM with Two NIC Cards – Existing Virtual Network



8. Fill or modify the following parameters:

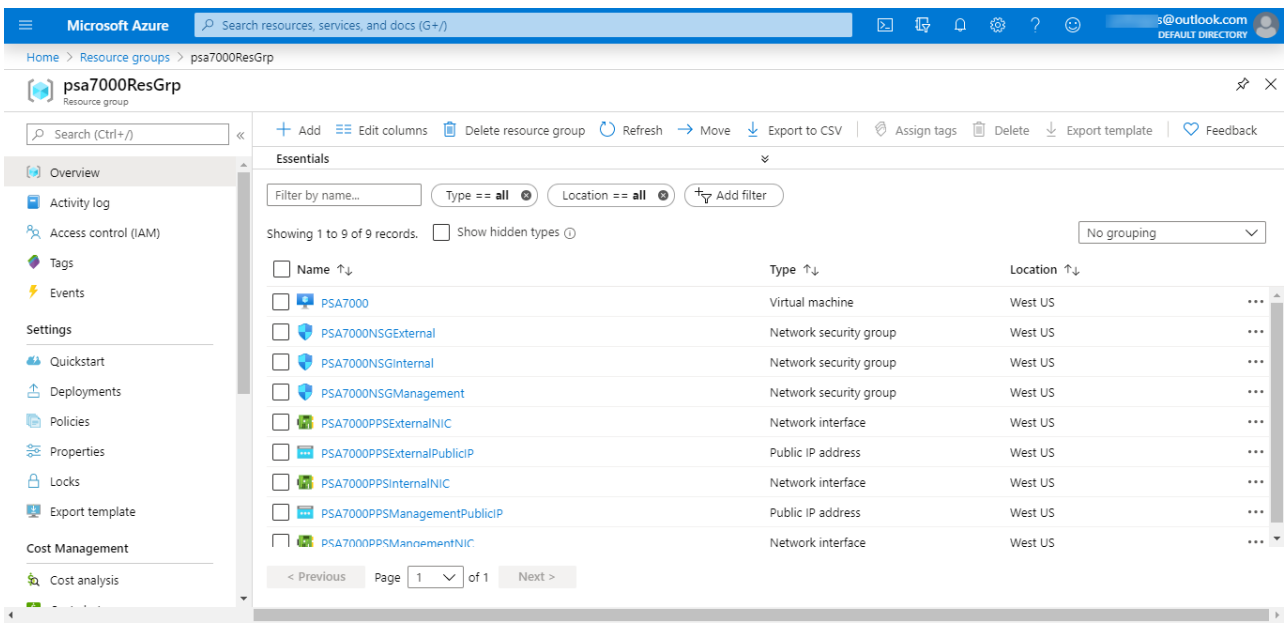
- **Resource group:** Specify the resource group name in which Pulse Policy Secure needs to be deployed
- **Location:** Region where resource group needs to be created
- **PPS Storage Account Name:** Storage account name where the Pulse Policy Secure Virtual Appliance image is available
- **PPS Storage Account Resource Group:** Resource group of where the Pulse Policy Secure Virtual Appliance image is copied
- **PPS Image Location URI:** URI to Pulse Policy Secure Virtual Appliance Image
- **PPS VM Name:** Name of the Pulse Policy Secure Virtual instance
- **PPS Config:** Provisioning parameters in XML format.
- **SSH Public Key:** This key is used to access PPS via SSH. The SSH keys are generated using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer:

For Windows: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows
For MacOS and Linux: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/mac-create-ssh-keys

- **DNS Label Prefix Int:** Prefix for the internal interface DNS label
- **Resource Group Name of Existing Virtual Network:** Resource Group name of the Virtual network
- **Existing Vnet Name:** Virtual network name
- **Existing Internal Subnet:** Subnet from which the Pulse Policy Secure internal interface needs to lease IP
- **Existing External Subnet:** Subnet from which the Pulse Policy Secure external interface needs to lease IP

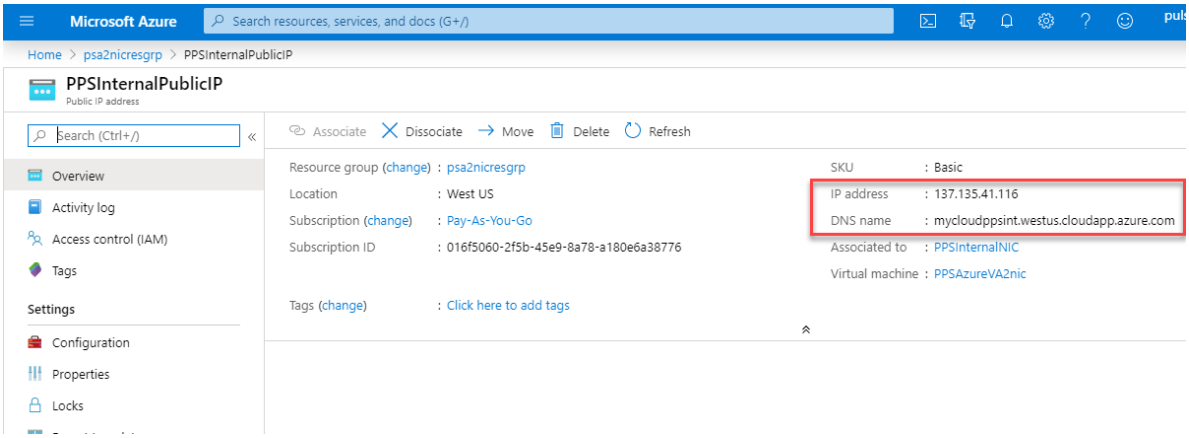
9. Agree to the Azure licensing terms and click **Purchase**.
10. Watch for the deployment succeeded message after 3 to 5 minutes.

Figure 27: Deployment Succeeded



11. Go to the resource group in which the Pulse Policy Secure Virtual appliance was deployed to see the resources created.
12. Click **PPS Internal Public IP** and note down the **PPS Internal Public IP and DNS name (FQDN)** to access PPS for end user page.

Figure 28: PULSE POLICY SECURE Internal Public IP



Note: Azure allows static as well as dynamic assignment of the IP addresses to the network interfaces. The mode of IP assignment (static/dynamic) can be mentioned in the Azure Resource Manage template file. The current JSON template uses dynamic method of allotting IP addresses to the network interfaces.

Deploying Pulse Policy Secure on Azure using Azure CLI

Before proceeding with the deployment, refer Upload Pulse Policy Secure Virtual Appliance Image to Azure.

1. Download and Install Azure CLI from https://azure.github.io/projects/clis.
2. Visit www.pulsesecure.net and download the **ps-pps-azure-psa-v-<releaseno>-<buildno>-package.zip** file.
3. Unzip the file and look for the **pulsesecure-pps-3-nics.zip** file. Unzip the file to get **azuredeploy.json**
4. Ensure that parameters section has correct default values:
   - **PPS Storage Account Name:** Storage account name where the Pulse Policy Secure Virtual Appliance image is available
   - **PPS Storage Account Resource Group:** Resource group where the Pulse Policy Secure Virtual Appliance image is copied
   - **PPS Image Location URI:** URI to the Pulse Policy Secure Virtual Appliance Image
   - **PPS VM Name:** Name of the Pulse Policy Secure Virtual instance
   - **PPS Config:** Provisioning parameters in an XML format. Refer "Pulse Policy Secure Provisioning Parameters"
   - **DNS Label Prefix Mgmt:** Prefix for the management interface DNS label
   - **Vnet Address Space:** Virtual network address space
   - **Internal Subnet:** Subnet from which the Pulse Policy Secure internal interface needs to lease IP
5. To deploy Pulse Policy Secure using Azure CLI, run the following commands

```
$ az login
$ az group create -l <location> -n <resource group name>
$ az group deployment create -g <resource group name> --template-file <json file name>
```

For example: C:\Users\xyz>az login

```
C:\Users\Sachin>az login
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "id": "87ca7834-b728-4432-b509-20f526eaada6",
    "isDefault": false,
    "name": "Pay-As-You-Go",
    "state": "Disabled",
    "tenantId": "d057727c-e52f-4a27-95a6-670c906a90be",
    "user": {
      "name": "pulsepps@outlook.com",
      "type": "user"
    }
  },
  {
    "cloudName": "AzureCloud",
    "id": "016f5060-2f5b-45e9-8a78-a180e6a38776",
    "isDefault": true,
    "name": "Pay-As-You-Go",
    "state": "Enabled",
    "tenantId": "d057727c-e52f-4a27-95a6-670c906a90be",
    "user": {
      "name": "pulsepps@outlook.com",
      "type": "user"
    }
  }
]

C:\Users\Sachin>
```

C:\Users\xyz>az group create -l westus -n MyResourceGroup

```
C:\Users\Sachin>az group create -l westus -n MyResourceGroup
{
  "id": "/subscriptions/016f5060-2f5b-45e9-8a78-a180e6a38776/resourceGroups/MyResourceGroup",
  "location": "westus",
  "managedBy": null,
  "name": "MyResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}

C:\Users\Sachin>
```

C:\Users\xyz\Downloads>az group deployment create -g MyResourceGroup --template-file forcli.json

```
      "id": null,
      "namespace": "Microsoft.Compute",
      "registrationPolicy": null,
      "registrationState": null,
      "resourceTypes": [
        {
          "aliases": null,
          "apiVersions": null,
          "capabilities": null,
          "locations": [
            "westus"
          ],
          "properties": null,
          "resourceType": "virtualMachines"
        }
      ]
    }
  ],
  "provisioningState": "Succeeded",
  "template": null,
  "templateHash": "6964255069003467159",
  "templateLink": null,
  "timestamp": "2019-12-16T10:39:19.300522+00:00"
},
"resourceGroup": "MyResourceGroup",
"type": "Microsoft.Resources/deployments"
}

C:\Users\Sachin\Downloads>
```

Pulse Policy Secure Provisioning Parameters

Provisioning parameters are those parameters which are required during the deployment of a virtual appliance. Pulse Policy Secure accepts the following parameters as provisioning parameters in the XML format.

```
"<pulse-config>
 <primary-dns>8.8.8.8</primary-dns>
 <secondary-dns>8.8.8.9</secondary-dns>
 <wins-server>1.1.1.1</wins-server>
 <dns-domain>psecure.net</dns-domain>
 <admin-username>admin</admin-username>
 <admin-password>password</admin-password>
 <cert-common-name>va1.psecure.net</cert-common-name>
 <cert-random-text>fdsfpisonvsfnms</cert-random-text>
 <cert-organisation>Psecure Org</cert-organisation>
 <config-download-url><value></config-download-url>
 <config-data><value></config-data>
 <auth-code-license><value></auth-code-license>
 <enable-license-server>n</enable-license-server>
 <accept-license-agreement>n</accept-license-agreement>
 <enable-rest>n</enable-rest>
</pulse-config>",
```

The below table depicts the details of xml file.

#	Parameter Name	Type	Description
1	primary-dns	IP address	Primary DNS for Pulse Policy Secure
2	secondary-dns	IP address	Secondary DNS for Pulse Policy Secure
3	wins-server	IP address	Wins server for Pulse Policy Secure
4	dns-domain	string	DNS domain of Pulse Policy Secure
5	admin-username	string	admin UI user name
6	admin-password	string	admin UI password
7	cert-common-name	string	Common name for the self-signed certificate generation. This certificate is used as the device certificate of Pulse Policy Secure Random text for the self-certificate generation Organization name for the self-signed certificate generation
8	cert-random-text	string	
9	cert-organization	string	
10	config-download-url	String URL	Http based URL where XML based Pulse Policy Secure configuration can be found. During provisioning, Pulse Policy Secure fetches this file and comes up with preloaded configuration. XML based configuration can be present in another VM in Azure cloud or at corporate network which is accessible for Pulse Policy Secure through site to site VPN between Azure and corporate data center
11	config-data	string	base64 encoded XML based Pulse Policy Secure configuration
12	auth-code-license	string	Authentication code that needs to be obtained from Pulse Secure
13	enable-license-server	string	If set to 'y', PPS will be deployed as a License server. If set to 'n', PPS will be deployed as a normal

			server.
14	accept-license-agreement	string	This value is passed to the instance for configuration at the boot time. By default, this value is set to 'n'. This value must be set to 'y'.
15	enable-rest	string	If set to 'y', REST API access for the administrator user is enabled.

Note: In the above list of parameters, **primary dns, dns domain, admin username, admin password, cert-random name, cert-random text, cert-organization** and **accept-license-agreement** are mandatory parameters. The other parameters are optional parameters.

Provisioning Pulse Policy Secure with Predefined Configuration

The Pulse Policy Secure Virtual Appliance can be provisioned on Azure with a pre-defined Pulse Policy Secure configuration. The provisioning can be done in the following two ways:

- Pulse Policy Secure administrator needs to provide the location of the XML-based configuration as a provisioning parameter. Refer 'Pulse Policy Secure Provisioning Parameters' for details about the Pulse Policy Secure specific provisioning parameters.

  Pulse Policy Secure configuration can be kept on Azure VM or on a machine located in the corporate network. If it is in the corporate network, the Pulse Policy Secure administrator needs to ensure that site-to-site VPN between Azure to corporate network is already established so that Pulse Policy Secure can access the machine located in the corporate network.
- Pulse Policy Secure administrator provides the configuration data encoded in the base64 encoded xml in the ARM template.

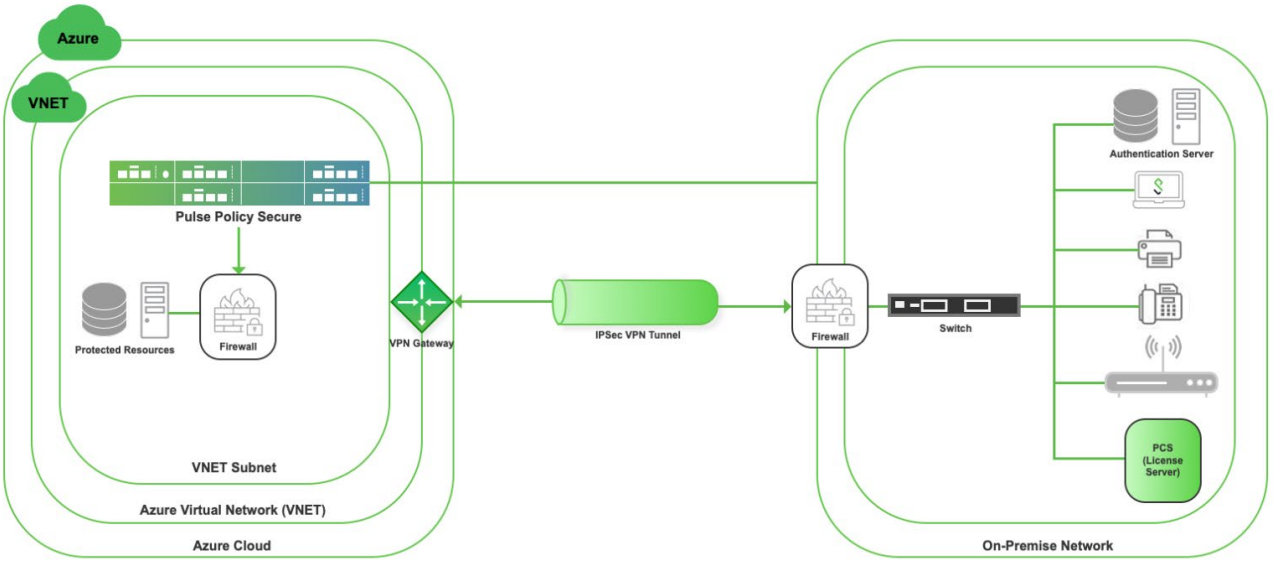
Configuring Licenses on the Pulse Policy Secure Appliance

By default, two-user licenses are provided. To add more licenses, the Pulse Policy Secure administrator needs to leverage the Pulse License server.
The Pulse License server can be made available in:

- corporate network
- cloud network

Pulse License Server in Corporate Network

Figure 29: Pulse License Server in a Corporate Network

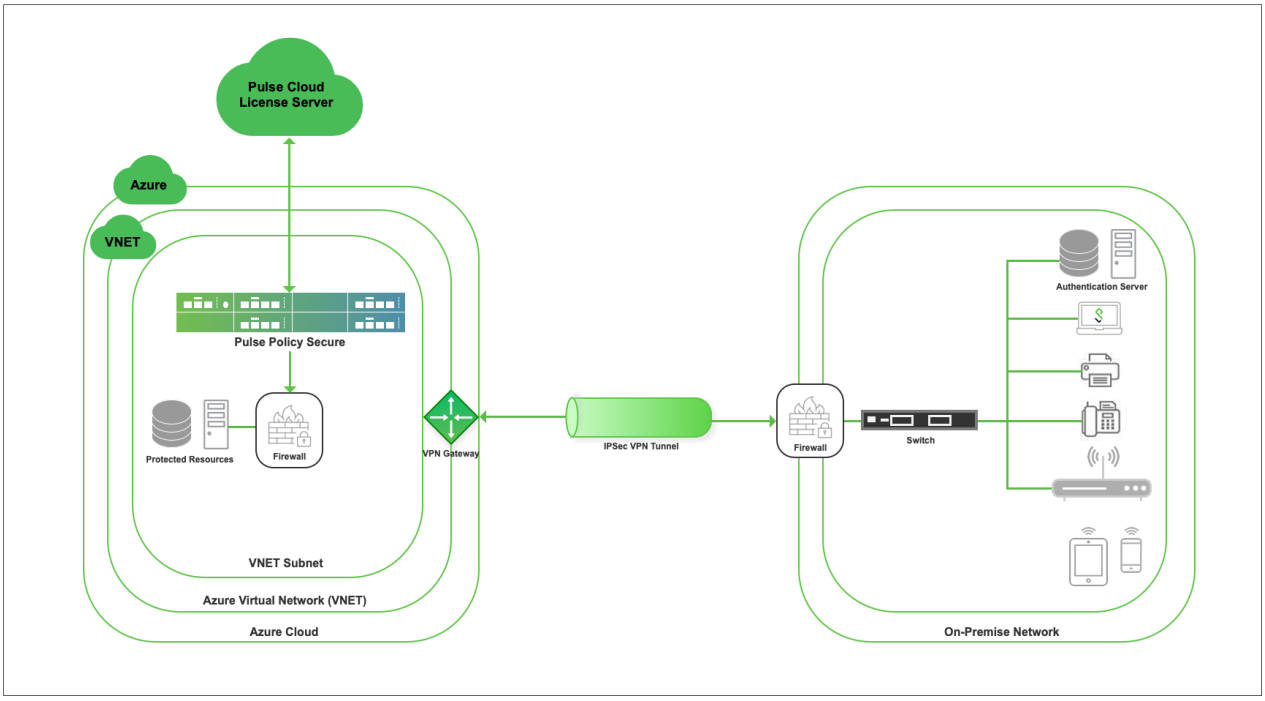


Pulse License Server in Cloud Network

Pulse Policy Secure virtual machines (VM) are enabled to provision licenses through the Pulse Cloud Licensing Service (PCLS). For this, administrator needs to obtain an Authentication code from Pulse Secure Support and apply it in Download Licenses page of PPS admin console. The PPS also periodically sends heartbeat messages to PCLS for auditing purposes.
The Authentication code can also be specified in the ARM template. When PPS comes up, it automatically fetches the Authentication code.

- Adding Authentication Code in PPS Admin Console
- Including Authentication Code in ARM Template

Figure 30: Pulse License Server in Cloud Network

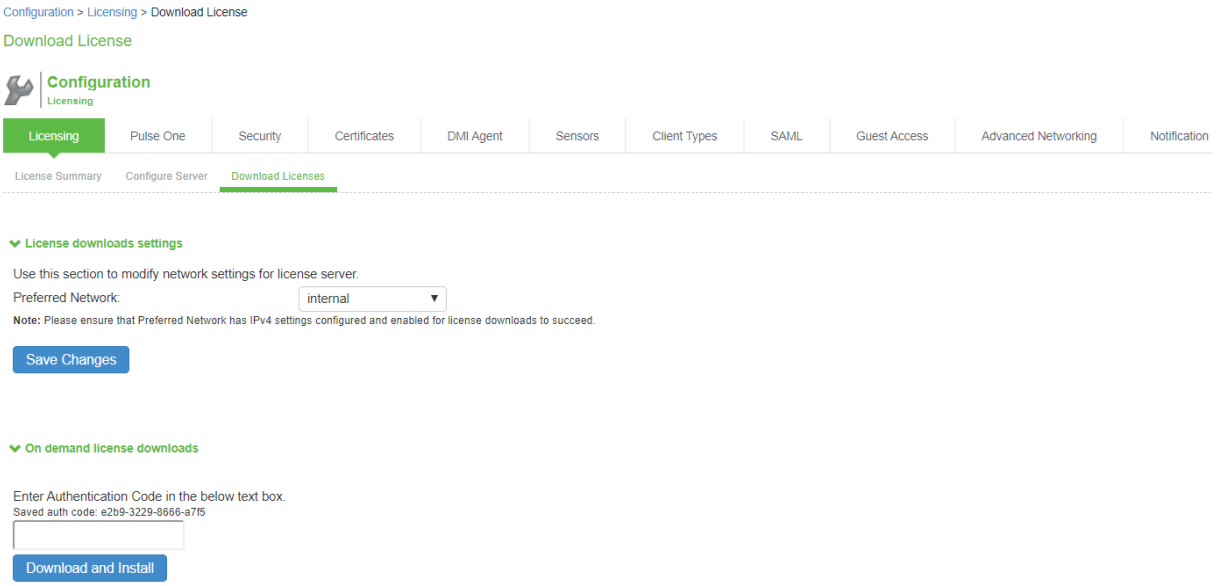


Adding Authentication Code in PPS Admin Console

To add Authentication code:

1. Go to **System > Configuration > Licensing > Download Licenses.**
2. Under On demand license downloads, enter the Authentication code in the text box.
3. Click on **Download and Install**.

Figure 31: Enter Authentication Code



Including Authentication Code in ARM Template

To include Authentication code in the ARM template:

1. In the ARM template, go to the PPSConfig section.
2. For the element <auth-code-license>, enter the Authentication code as the content.
3. Save the template.

```
"defaultValue":
"<pulse-config><primary-dns>8.8.8.8</primary-dns><secondary-dns>8.8.8.9</secondary-dns><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net</dns-dom
ain><admin-username>admin</admin-username><admin-password>password</admin-password><cert-common-name>va1.psecure.net</cert-common-name><cert-random-text>
fdsfpisonvsfnms</cert-random-text><cert-organisation>Psecure
Org</cert-organisation><config-download-url></config-download-url><config-data></config-data><auth-code-license></auth-code-license><enable-license-serve
r>n</enable-license-server><accept-license-agreement>n</accept-license-agreement></pulse-config>",
```

For details about the license configuration, refer to License Configuration Guide.

Accessing the Pulse Policy Secure Virtual Appliance

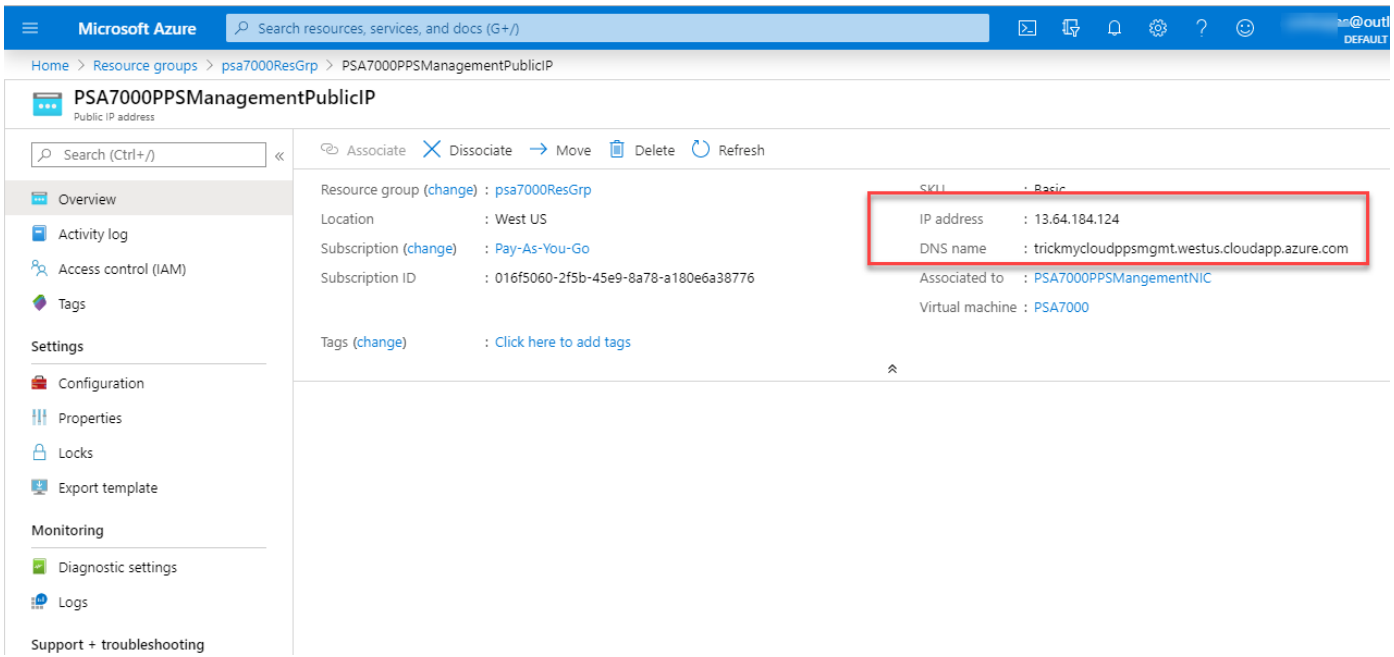
The Pulse Policy Secure appliance can be accessed:

- as an administrator
- as an end user
- using SSH console

Accessing the Pulse Policy Secure Virtual Appliance as an Administrator

To access the Pulse Policy Secure Virtual Appliance as an administrator, copy the IP address from the Pulse Management Interface resource.

Figure 32: Pulse Management Interface



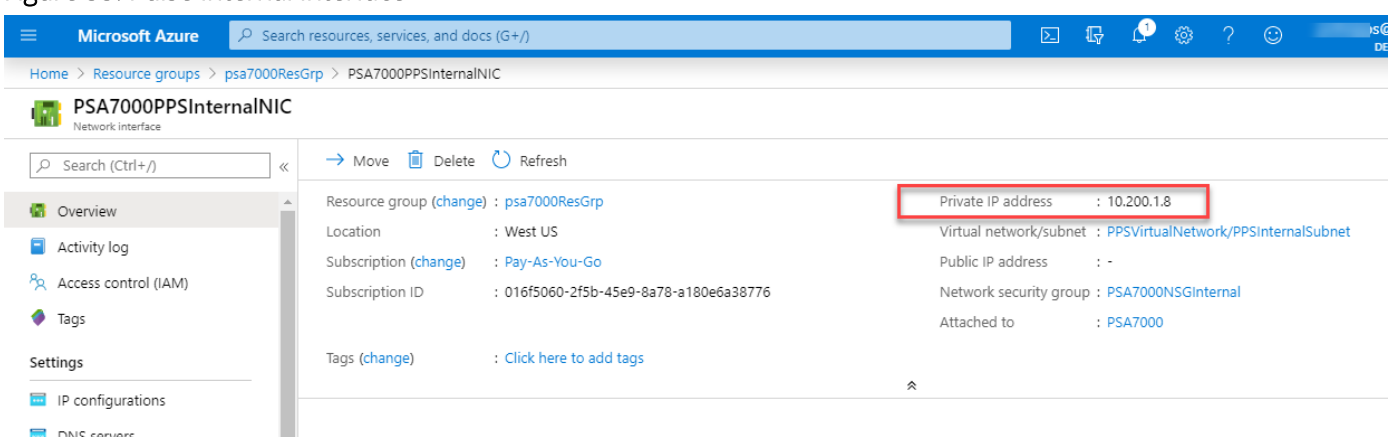
Use the credentials provided in the provisioning parameters to log in as the administrator. The default PPS admin UI user configured in the azuredeploy.json config file is: user 'admin' and password 'password1234'.

The administrator can configure Active Directory located in the corporate network for user authentication. The Pulse Policy Secure Virtual Appliance administrator can check troubleshooting tools provided in the Pulse Policy Secure admin UI (System->Maintenance->Troubleshooting), to verify whether Pulse Policy Secure is able to reach other cloud resources as well as corporate resources.

Accessing the Pulse Policy Secure Virtual Appliance as an End User

To access the Pulse Policy Secure Virtual Appliance as an end user, copy the Private IP address from Pulse Internal Interface resource.

Figure 33: Pulse Internal Interface



Accessing the Pulse Policy Secure Virtual Appliance using SSH Console

To access the Pulse Policy Secure Virtual Appliance using the SSH console, copy the Public IP address from the PPSManagementPublicIP resource.

On Linux and Mac OSX

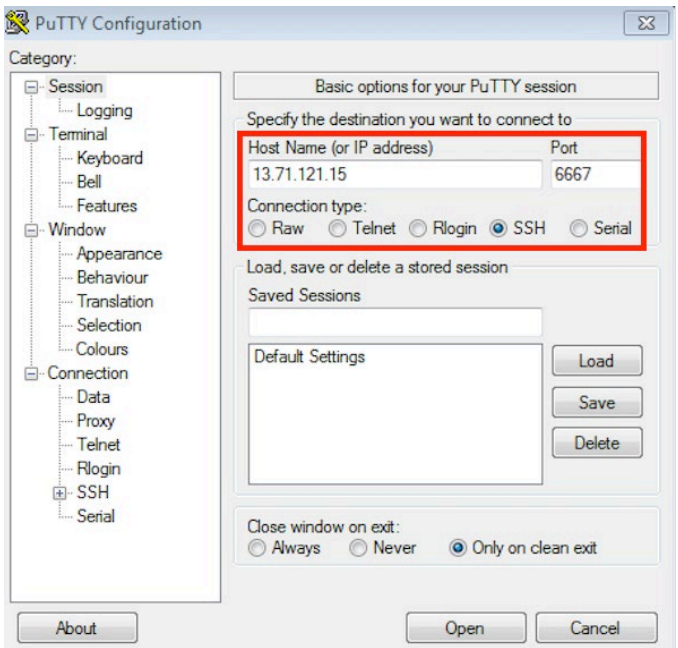
Execute the following command:

```
ssh -i <rsa-private-key-file> <PPS-Management-Interface-PublicIP> -p 6667
```

On Windows

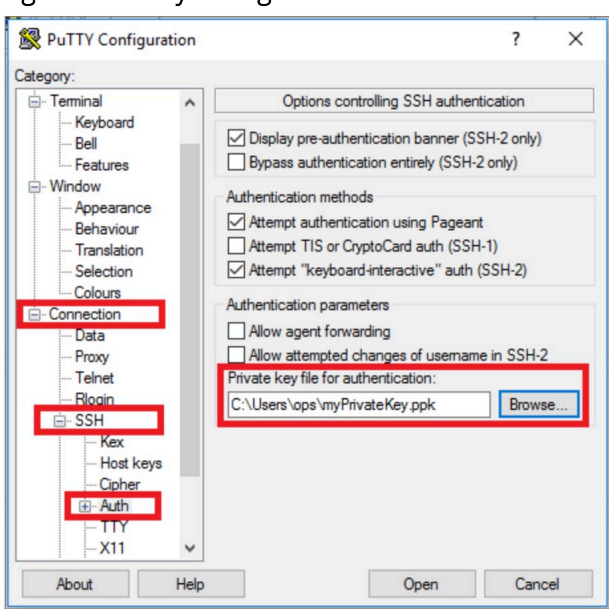
1. Launch the Putty terminal emulator.
2. In the Session category:
   - Enter the host name or IP address.
   - Enter the port number.
   - Select the connection type as SSH.

Figure 34: Putty Configuration – Basic Options



3. Select **Connection > SSH > Auth**. Click **Browse** and select the private key file for authentication.

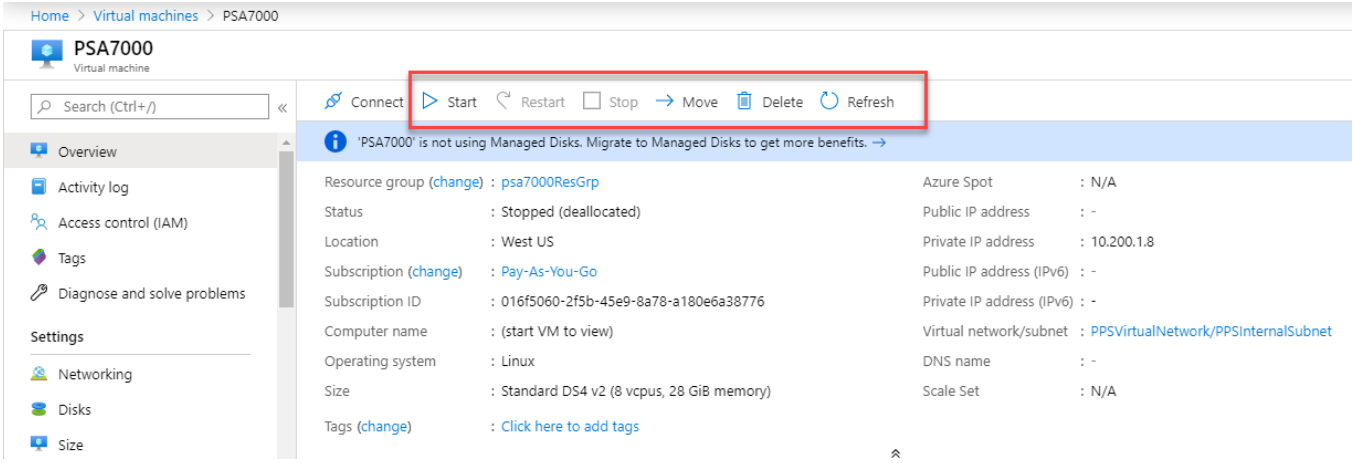
Figure 35: Putty Configuration – SSH Authentication



System Operations

The Azure VA portal provides Start, Restart and Stop operations to control the Virtual Appliance connection.

Figure 36: System Operations



On the Azure portal top menu bar:

- Click **Start** to start a VM
- Click **Stop** to stop the VM
- Click **Restart** to restart the VM

The corresponding CLI commands are:

Start a VM

```
az vm start --resource-group myResourceGroup --name myVM
```

Stop a VM

```
az vm stop --resource-group myResourceGroup --name myVM
```

Restart a VM

```
az vm restart --resource-group myResourceGroup --name myVM
```

Network Configuration

IP Address Assignment for Internal, External and Management Interfaces

Each interface in Azure can have private and public IP addresses. Sample Azure 3NIC Templates provided by Pulse Policy Secure creates the Pulse Policy Secure Virtual Appliance with public and private IP addresses for management interfaces and only private IP address for internal and external interface Sample Azure 2NIC Templates provided by Pulse Policy Secure creates the Pulse Policy Secure Virtual Appliance with public and private IP addresses for internal interface and only private IP address for external interface More details about IP address types on Azure can be seen at: https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-ip-addresses-overview-arm

IP Addressing Modes

When Pulse Policy Secure gets deployed by using the sample templates provided by Pulse Secure, Pulse Policy Secure comes up with multiple interfaces. If you take an example of a template “pulsesecure-pps-3-nics.zip” provided by Pulse Secure, you notice the following things.

PPS management interface is having both Public and Private IP addresses. In the below code snippet, observe the network interface getting created with two IP addresses - private IP address and public IP address. Highlighted section points to private IP allocation method and Public IP address getting assigned to NIC.

```
"type": "Microsoft.Network/networkInterfaces",
"name": "[variables('ppsExtNic')]",
 ----------
 ----------
"properties": {
 "privateIPAllocationMethod": "Dynamic",
"privateIPAddressVersion": "IPv4",
"publicIPAddress": {
"id": "[resourceId('Microsoft.Network/publicIPAddresses',variables('publicIPAddr1'))]"
},
```

If you want to have control on the IP assigned to Network Interface, then you need to change the attribute “privateIPAllocationMethod” from “Dynamic” to “Static”. Also, you need to add an attribute called “privateIPAddress” which holds the static IP address in the variables section. When you are assigning static IP address, make sure that it is not in the reserved IP category.

```
"ipConfigurations": [{
   "name": "ipconfig2",
   "properties": {
      "privateIPAllocationMethod": "Static",
      "privateIPAddressVersion": "IPv4",
      "privateIPAddress": "[variables('privateIPExternal')]",
   }
}]
```

Modifying Network Parameters After Deployment

Since Networking Infrastructure is provided by Azure, a PPS admin cannot change Networking configuration after deployment. Hence, both admin UI and ssh does not support changing network configuration.

Controlling the Selection of Internal, External and Management Interfaces

Sample Azure Template, provided by Pulse Secure, requests Azure fabric to create three Network Interfaces. While running this template, Azure fabric creates interfaces named eth0, eth1 and eth2 and attaches them to PPS Virtual Interface.
So, the question is, among eth0, eth1 and eth2 which network interface will become external, internal or management interface? Below table answers this question.

Interface Name	PPS Interface
eth0	internal interface
eth1	external interface
eth2	management interface

Then, question is how you can control the order of network interfaces named eth0, eth1 and eth2 created through Azure Template?. Azure supports two types of interfaces: primary and secondary. Only one primary interface can be present on a VM.
For more details of primary and secondary interface, see https://docs.microsoft.com/en-us/azure/virtual-

network/virtual-network-network-interface-addresses.

The Pulse Policy Secure Virtual Appliance is qualified with internal interface as primary and other two are secondary. In the following code snippet, three network interfaces get assigned to VM. These three NICs with ID "nic1", "nic2" and "nic3" are internally mapped to 'eth0', 'eth1', and 'eth2' respectively.

```
"networkProfile": {
  "networkInterfaces": [{
    "id": "nic1",
    "properties": {
      "primary": true
    }
  }, {
    "id": "nic2",
    "properties": {
      "primary": false
    }
  }, {
    "id": "nic3",
    "properties": {
      "primary": false
    }
  }]
},
```

PPS converts eth0 to int0, eth1 to ext0 and eth2 to mgmt0. This means, the network interface with ID nic1 will be internal interface, nic2 will be external interface and nic3 will be management interface.
The below table depicts this scenario well:

Interface Name	PPS Interface	Network ID
eth0	internal interface (int0)	nic1
eth1	external interface (ext0)	nic2
eth2	management interface (mgmt0)	nic3

Suppose if you make 'nic2' as primary, then the order may not be maintained, and it is difficult to predict which interface will become internal interface of PPS. As a best practice, always assign 'primary' to the first network interface which will become internal interface of PPS.

Backing up Configs and Archived Logs on Azure Storage

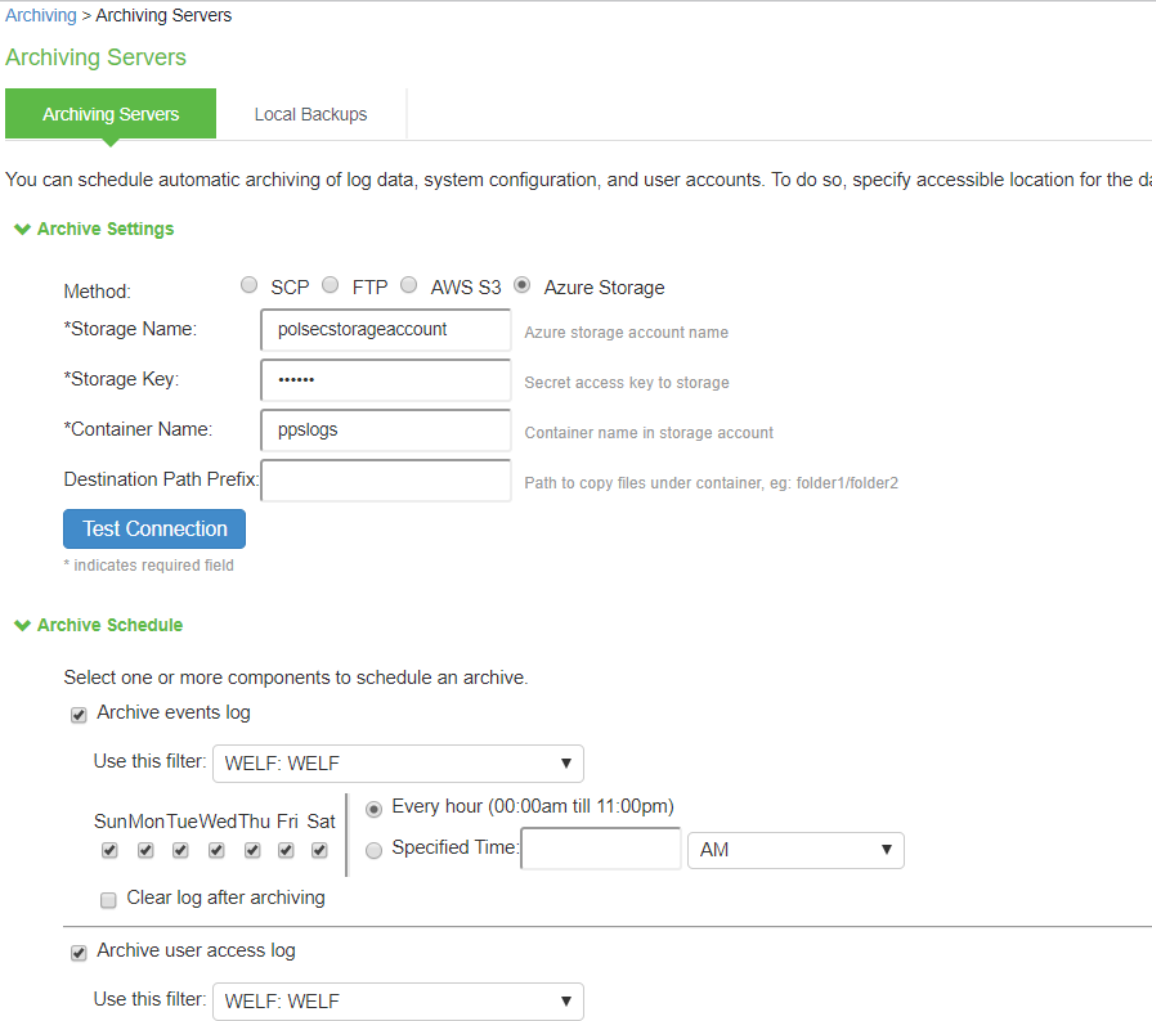
Pulse Policy Secure supports pushing configs and archived logs to the servers that support SCP and FTP protocols. In the Azure deployment, Pulse Policy Secure now supports pushing configs and archived logs to the Azure storage.

Configuring Backup Configs and Archived Logs via PPS Admin Console

To configure backing up configs and archived logs:

1. Log into the Pulse Policy Secure admin console.
2. Navigate to **Maintenance > Archiving > Archiving Servers**.
3. In the Archive Settings section, select the **Azure Storage** option and configure Storage Name, Storage Key, Container Name and Destination Path Prefix.

   Figure 37: Azure Archive Settings



Parameter	Description
Storage Name	To create an Azure V2 Storage account: 1. In the Azure portal, select **All services**. 2. From the list of resources, select **Storage Accounts**. 3. In the Storage Accounts window, click **Add**. 4. Select the subscription in which to create the storage account. 5. Under the **Resource group** field, select **Create new** and enter a name for the new resource group. 6. Next, enter a unique name, between 3 and 24 characters length, for the storage account. For the procedure to create storage account, refer https://docs.microsoft.com/en-us/azure/storage/common/storage-quickstart-create-account?tabs=azure-portal
Storage Key	To view storage key, 1. In the Azure portal, locate the storage account (see Storage Name description). 2. In the Settings section, select **Access keys**. The account access keys and the complete connection string for each key appear. 3. Find the Key value under key1 and click the **Copy** button to copy the account key. For more details, refer https://docs.microsoft.com/en-us/azure/storage/common/storage-account-manage#view-and-copy-access-keys
Container Name	Container name in the storage account.

Dest Path Prefix (Optional)	Path to copy files under container.

Configuring Backup Configs and Archived Logs via REST

Setting Azure as Archive Logs Backup

```
REQUEST
PUT /api/v1/configuration/system/maintenance/archiving/settings HTTP/1.1
Content-Type: application/json
{
    "archive-path": "folder1/folder2",
    "method": "AZURE",
    "Password-cleartext": "fasfdsfsdasfas",
    "server": "mystorage",
    "user-name": "mycontainer"
}
```

Mapping of keys in POST body:

archive-path	Destination path Prefix
method	method (AZURE)
Password-cleartext	Storage Key
server	Storage Name
user-name	Container Name

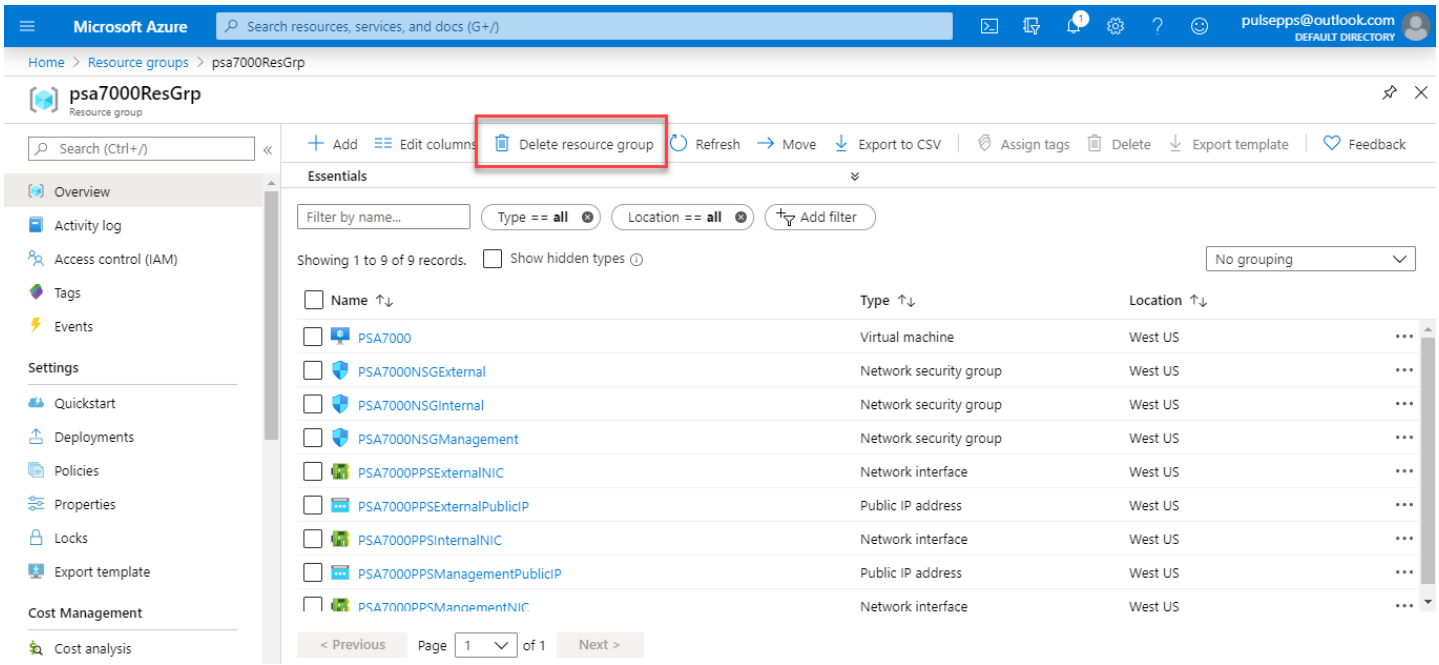
Decommissioning Pulse Policy Secure

When deploying Pulse Policy Secure, if you have selected the option “Use existing resource group”, then follow the steps mentioned in the section Delete Pulse Policy Secure and Resource It Uses, but not the Other Resources in Resource Group. Else if you have selected the option “New resource group” then follow the steps mentioned in the section Delete Entire Resource Group that the Pulse Policy Secure Is In.

Delete Entire Resource Group that the Pulse Policy Secure Is In

1. Log into Azure portal.
2. Navigate to Resource Groups.
3. Click on the resource group where Pulse Policy Secure is in.
4. Click on the **Delete resource group** button. In the confirmation page type in resource group name and click **Delete**.

Figure 38: Delete Resource Group



5. Navigate to the storage account where the Pulse Policy Secure VHD image is stored.
6. In the storage account, click on **Container**. Find boot diagnostic folder and delete it. Boot diagnostic folder name will have the pattern “bootdiagnostics-<pps-name>-<random-ascii-characters>”.
7. In the storage account, click on respective **Container**. Find and click on the **vhds** folder. Find and delete file size named “<pps-name><13 digit unique string>ppsOSDisk.vhd”.

Delete Pulse Policy Secure and Resource It Uses, but not the Other Resources in Resource Group

1. Log into Azure portal.
2. Navigate to Resource Groups.
3. Click on the resource group where Pulse Policy Secure is in.
4. Delete the following resources:
   - PPS Virtual Machine
   - Virtual Network named PPSVirtualNetwork
   - PPSInternalNIC, PPSExternalNIC and PPSManagementNIC
   - PPSExternalPublicIP and PPSManagementIP
   - Three Network Security Groups named NSGInternal, NSGExternal and NSGManagement
   - User-defined Routing table named Backend2PPSRoute
5. Navigate to the storage account where the Pulse Policy Secure VHD image is stored.
6. In the storage account, click on **Blobs**. Find boot diagnostic folder and delete it. Boot diagnostic folder name will have the pattern “bootdiagnostics-<pps-name>-<random-ascii-characters>”.
7. In the storage account, click on **Blobs**. Find and click on the **vhds** folder. Find and delete file size named “<pps-name><13-digit unique string>ppsOSDisk.vhd”.

Pricing

The cost of running this product is combination of License cost and Azure infrastructure cost. It will be very difficult to find out Azure infrastructure cost for this product as it may vary with Regions/Country/Time. Hence, it is recommended to use "Azure Calculator", which is available online, to calculate the cost of running this product. Here are resources that are created during deployment.

Resources	Category	Chargeable
PPS VM (Standard_DS3_V2)	Compute	Yes
Virtual Network with four subnets	Networking	No
Three NIC cards named PPSInternalNIC, PPSExternalNIC and PPSManagementNIC	Networking	No
Two static Public IPs name PPSExternalPublicIP and PPSManagementIP	Networking	Yes
Three Network Security Groups named NSGInternal, NSGExternal and NSGManagement.	Networking	No
Boot diagnostic file under existing storage account (Less than 5MB)	Storage	Yes
File size of 40GB in the existing storage account under Blobs and container VHDs named "<pps-name><13 digit unique string>ppsOSDisk.vhd"	Storage	Yes

Limitations

The following list of Pulse Policy Secure features are not supported in this release:

- VLAN tagging
- IPv6 capabilities
- Layer 2 functionality like ARP Cache and ND Cache
- Virtual Ports

Not Qualified

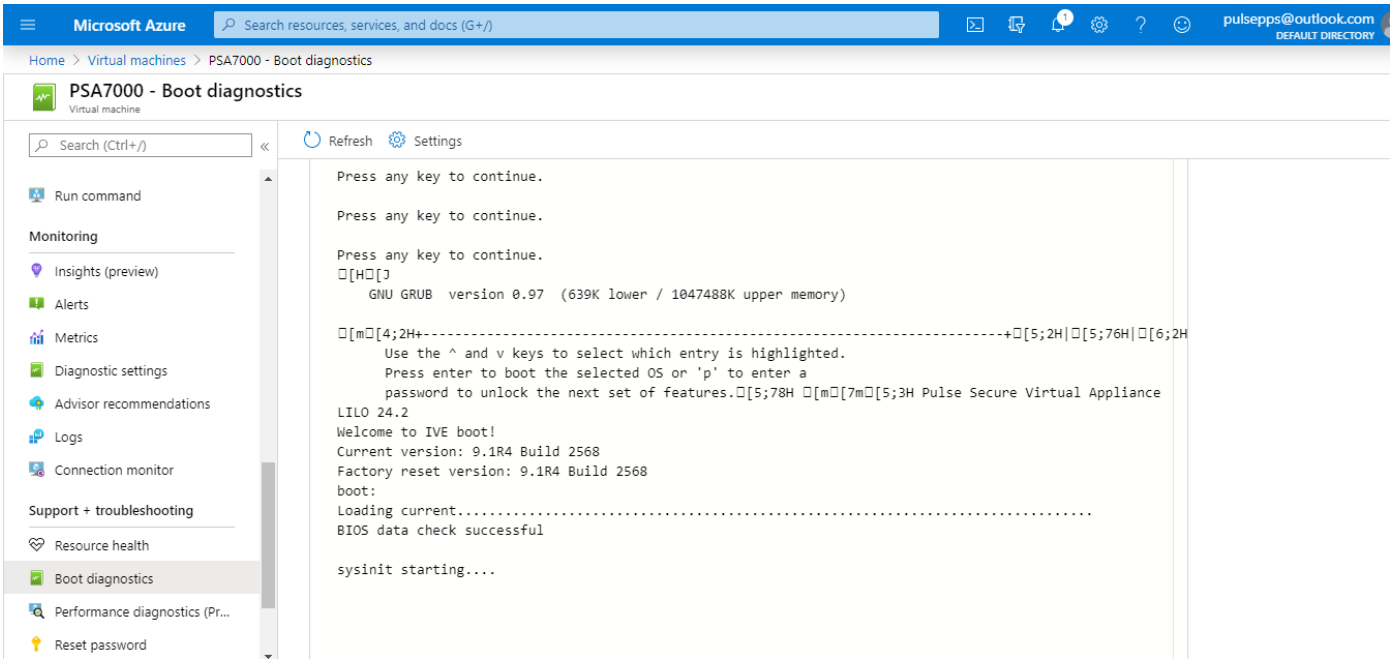
The following list of Pulse Policy Secure features are not qualified in this release:

- Pulse Policy Secure and Pulse One interaction
- IF-MAP support

Troubleshooting

Pulse Policy Secure emits booting logs at a specified storage. You can check the storage details of the boot diagnostic logs as shown below:

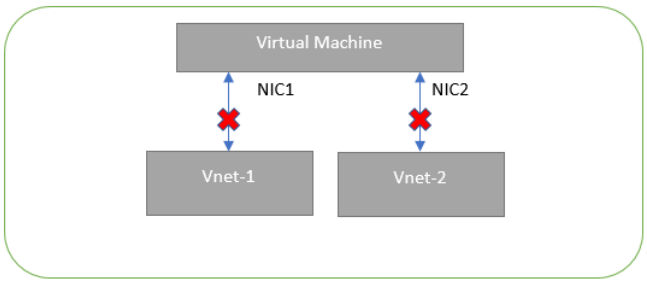
Figure 39: Boot Diagnostics



Appendix A: Network Security Group (NSG)

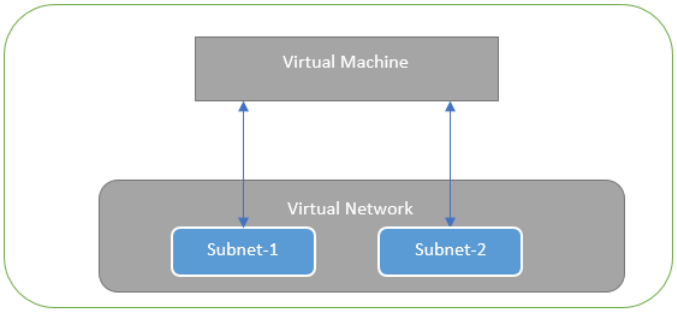
Microsoft Azure has a limitation where virtual machine with multiple network interfaces cannot connect to different Virtual Networks (VNETs). For example, a VM with two NIC cards, NIC1 and NIC2, will not be able to connect to Vnet1 and Vnet2 respectively.

Figure 40: Virtual Machine with two NIC cards Connecting to VNet1 and Vnet2



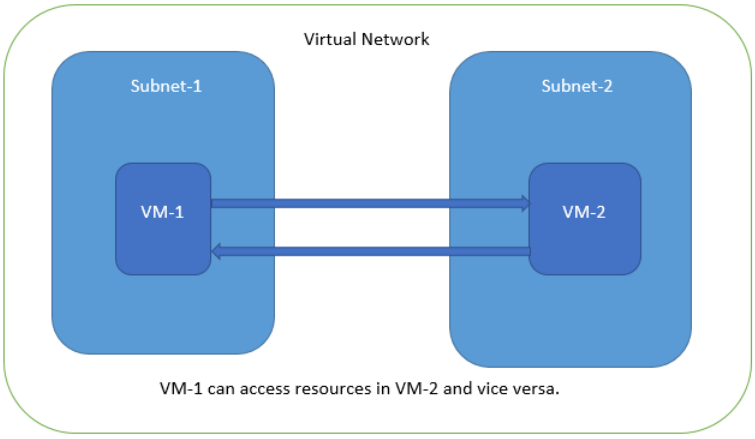
Microsoft Azure supports a virtual machine with multiple NICs to connect to different Subnets under a same Virtual Network. For example, a VM with two NICs, NIC1 and NIC2, can connect to 'Subnet1' and 'Subnet2' where these subnets exist under a same Virtual Network respectively.

Figure 41: Virtual Machine with two NICs Connecting to Subnet1 and Subnet2



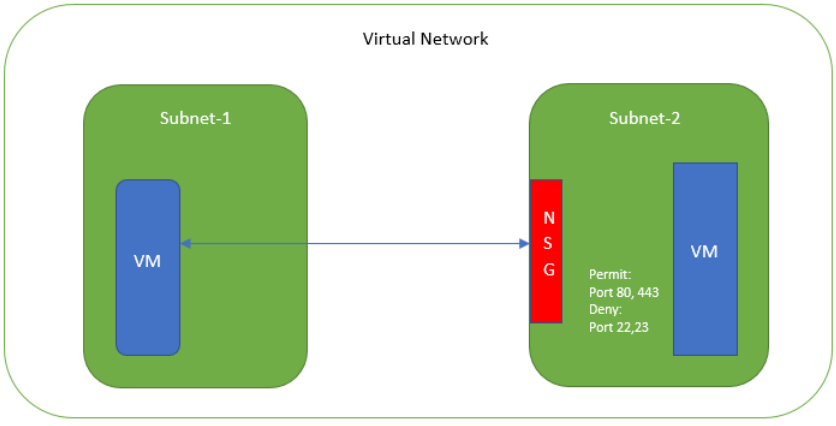
Azure provides isolation between different Vnets. But it does not provide the same kind of isolation when it comes to subnets in the same Vnet. For example, consider a Vnet has two subnets, Subnet1 and Subnet2. And consider two VMs, VM-1 and VM-2, which are connected to Subnet1 and Subnet2 respectively. In this scenario VM-1 can access the resources from VM-2 and vice versa.

Figure 42: Virtual Machine VM-1 can Access Resources in VM-2 and Vice Versa



Application isolation is an important concern in enterprise environments, as enterprise customers seek to protect various environments from unauthorized or unwanted access. To achieve the traffic isolation between subnets, go for an option of filtering traffic using “Network Security Group” provided by Azure.

Figure 43: Traffic Filtering by MS Azure Network Support Group



Pulse Policy Secure, when provisioned through the ARM template provided by Pulse Secure, creates three subnets under a virtual network named “PPSVirtualNetwork”.
The Subnets are:

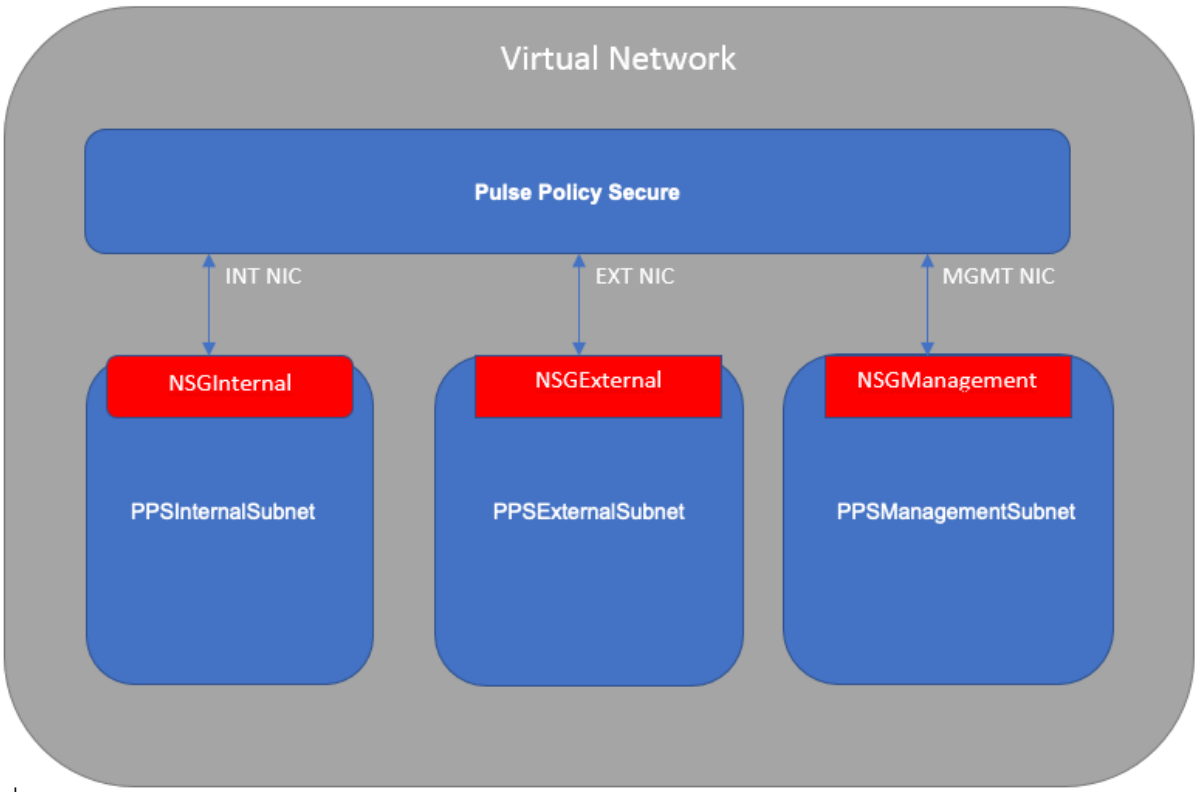
1. PPSInternalSubnet
2. PPSExternalSubnet

3. PPSManagementSubnet

Along with above mentioned subnets, create the following three Network Security Groups (NSG) policies:

1. NSGExternalSubnet
2. NSGInternalSubnet
3. NSGManagementSubnet

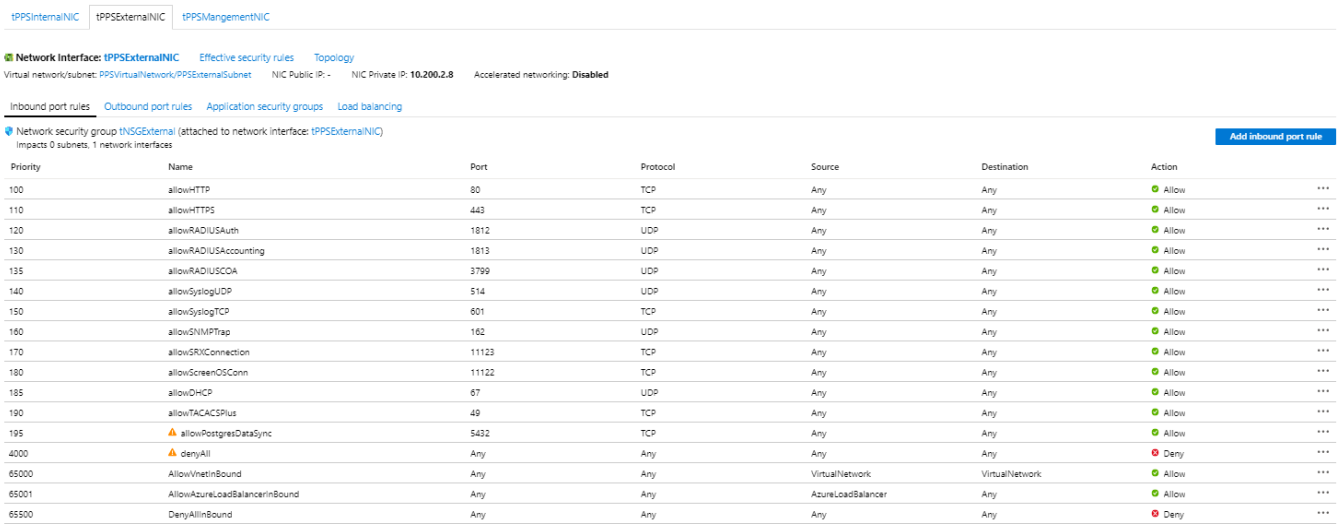
Figure 44: NSG External, Internal and Management Subnets



In Network Security Group (NSG) we need to create policies for Inbound and outbound traffic.

1. The list of NSG Inbound/Outbound rules created "NSGExternalSubnet" are:

Figure 45: NSG External - Inbound Rules



tPPSInternalNIC | tPPSExternalNIC | tPPSMangementNIC

Network Interface: tPPSExternalNIC Effective security rules Topology

Virtual network/subnet: PPSVirtualNetwork/PPSExternalSubnet NIC Public IP: - NIC Private IP: **10.200.2.8** Accelerated networking: **Disabled**

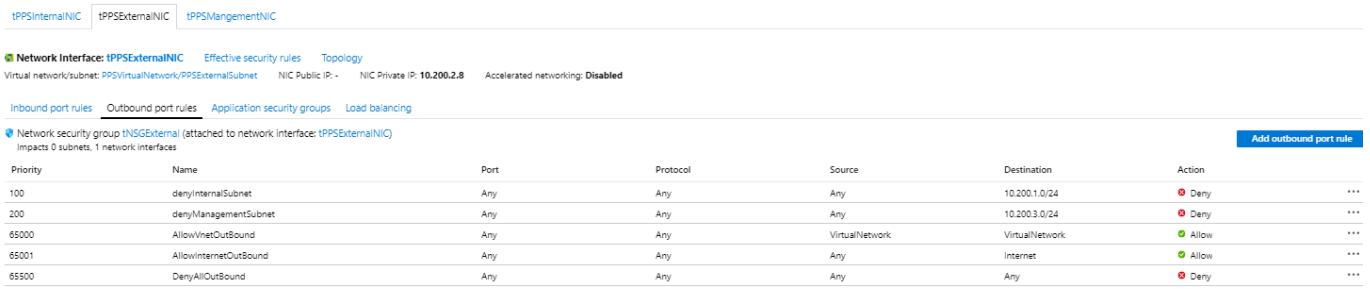
Inbound port rules | Outbound port rules | Application security groups | Load balancing

Network security group tNSGExternal (attached to network interface: tPPSExternalNIC)
Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
100	allowHTTP	80	TCP	Any	Any	Allow
110	allowHTTPS	443	TCP	Any	Any	Allow
120	allowRADIUSAuth	1812	UDP	Any	Any	Allow
130	allowRADIUSAccounting	1813	UDP	Any	Any	Allow
135	allowRADIUSCOA	3799	UDP	Any	Any	Allow
140	allowSyslogUDP	514	UDP	Any	Any	Allow
150	allowSyslogTCP	601	TCP	Any	Any	Allow
160	allowSNMPTrap	162	UDP	Any	Any	Allow
170	allowSRXConnection	11123	TCP	Any	Any	Allow
180	allowScreenOSConn	11122	TCP	Any	Any	Allow
185	allowDHCP	67	UDP	Any	Any	Allow
190	allowTACACSPlus	49	TCP	Any	Any	Allow
195	allowPostgresDataSync	5432	TCP	Any	Any	Allow
4000	denyAll	Any	Any	Any	Any	Deny
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Figure 46: NSG External - Outbound Rules



2. The list of NSG Inbound/Outbound rules created "**NSGInternalSubnet**" are:

Figure 47: NSG Internal - Inbound Rules

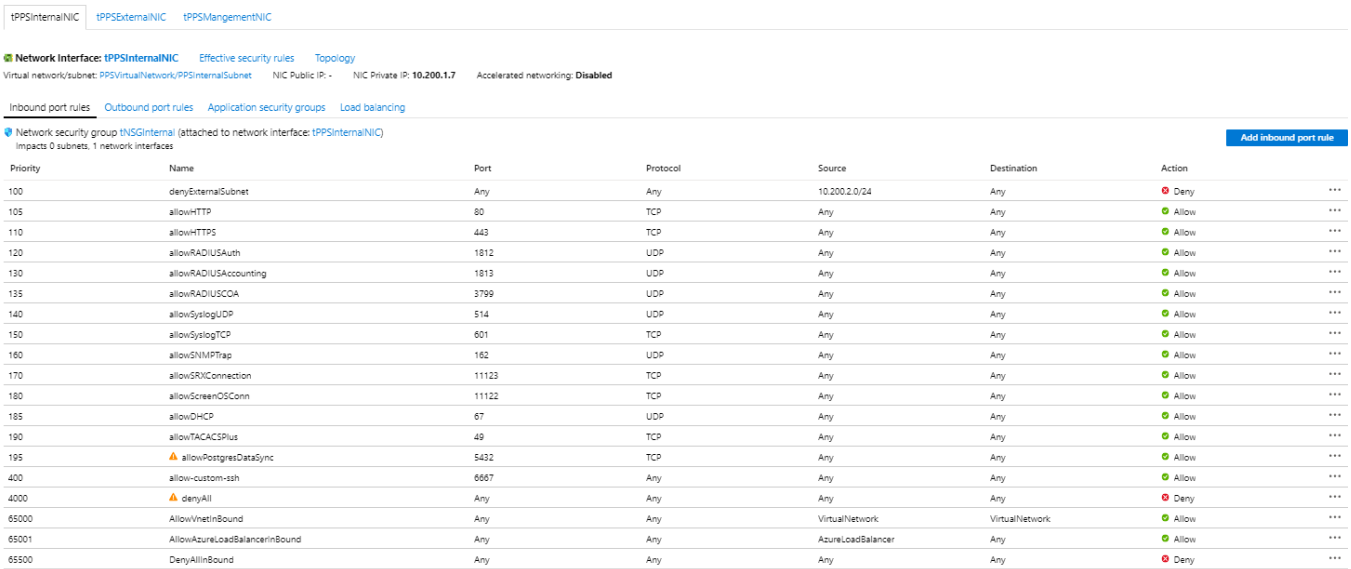
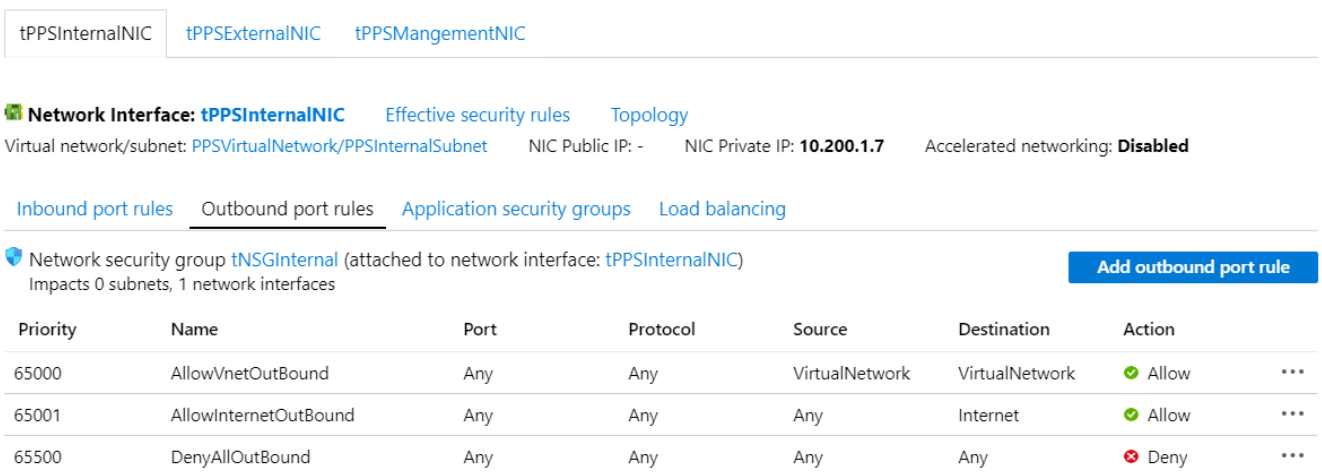


Figure 48: NSG Internal - Outbound Rules



3. The list of NSG Inbound/Outbound rules created "**NSGManagementSubnet**" are:

Figure 49: NSG Management - Inbound Rules

tPPSInternalNIC tPPSExternalNIC tPPSMangementNIC

Network Interface: tPPSMangementNIC Effective security rules Topology

Virtual network/subnet: PPSVirtualNetwork/PPSManagementSubnet NIC Public IP: **13.91.130.217** NIC Private IP: **10.200.3.4** Accelerated networking: **Disabled**

Inbound port rules Outbound port rules Application security groups Load balancing

Network security group tNSGManagement (attached to network interface: tPPSMangementNIC)
Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
100	allowHTTPS	443	TCP	Any	Any	Allow
200	allowHTTP	80	TCP	Any	Any	Allow
300	allowCustomSSH	6667	Any	Any	Any	Allow
400	allowDMI	830	Any	Any	Any	Allow
4000	denyAll	Any	Any	Any	Any	Deny
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Figure 50: NSG Management - Outbound Rules

tPPSInternalNIC tPPSExternalNIC tPPSMangementNIC

Network Interface: tPPSMangementNIC Effective security rules Topology

Virtual network/subnet: PPSVirtualNetwork/PPSManagementSubnet NIC Public IP: **13.91.130.217** NIC Private IP: **10.200.3.4** Accelerated networking: **Disabled**

Inbound port rules Outbound port rules Application security groups Load balancing

Network security group tNSGManagement (attached to network interface: tPPSMangementNIC)
Impacts 0 subnets, 1 network interfaces

Add outbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
100	denyInternalSubnet	Any	Any	Any	10.200.1.0/24	Deny
200	denyExternalSubnet	Any	Any	Any	10.200.2.0/24	Deny
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

Appendix B: Pulse Policy Secure Resource Manager Template

Pulse Secure provides sample Azure template files to deploy the Pulse Policy Secure Virtual Appliance on Azure. Users can modify this to make it suitable for their need. Visit https://www.pulsesecure.net and download the pulsesecure-pps-3-nics.zip file, and unzip it to get **azuredeploy.json**.

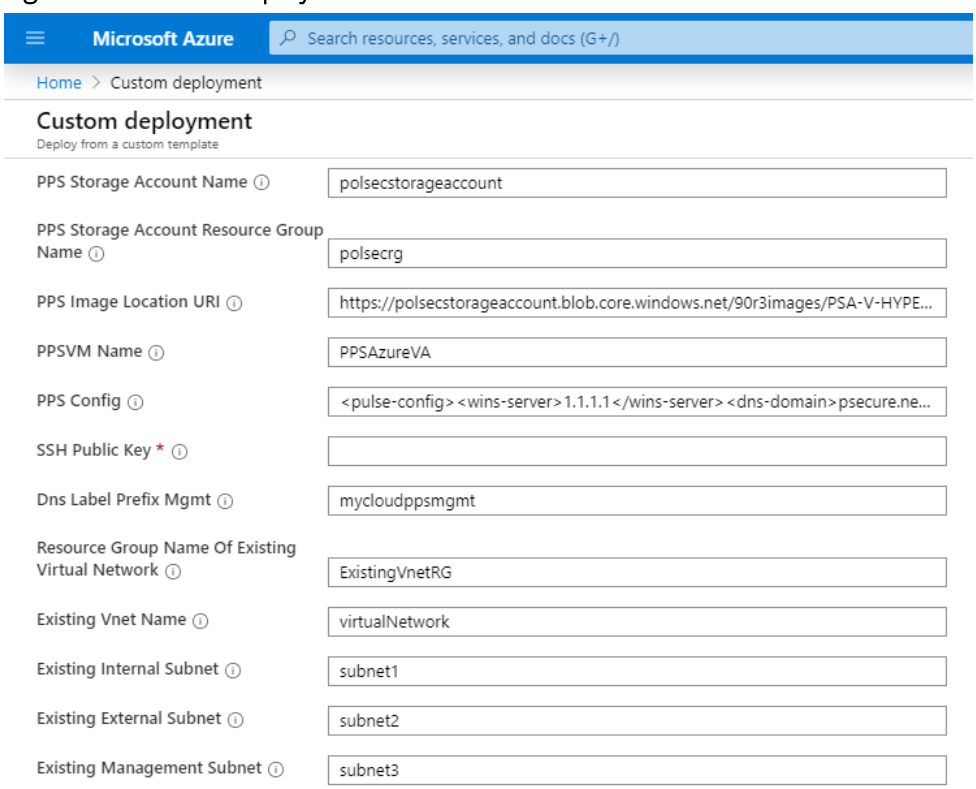
This template creates a new PPS with 3 NICs, Vnet, three subnets, NSG policies attached to PPS internal, external and management subnets. All 3 NICs of PPS are configured with dynamic IP configuration and enabled IP forwarding. Public IPs are attached to the PPS external and management NIC.

The template has following sections:

parameters	This section defines the parameters used for deploying PPS on Azure. It contains parameter name, its default value and the mouse-over help text that is displayed when mouse is placed over the parameter in Azure Web portal. The parameters defined here are displayed in the Custom Deployment page of Azure portal.
variables	This section defines variables that will be used in the functions defined in the resources section.
resources	This section defines resource types that are deployed or updated in a resource group.
outputs	This section defines the public IP address and FQDN returned after successful deployment of PPS on Azure.

parameters

Figure 51: Custom Deployment



PPS Storage Account Name: This is the name of the PPS Storage Account where the PPS Azure vhd image is stored.

```
"parameters": {
    "PPSStorageAccountName": {
    "type": "string",
    "defaultValue": "polsecstorageaccount",
    "metadata": {
      "description": "Storage account name where PPS image is uploaded"
    }
  },
```

PPS Storage Account Resource Group Name: The is the name of the PPS Storage Account Resource Group where the PPS Azure vhd image is stored.

```
"PPSStorageAccountResourceGroupName": {
  "type": "string",
  "defaultValue": "polsecrg",
  "metadata": {
    "description": "Resource group of the existing storage account where PPS image is uploaded"
  }
},
```

PPS Image Location URI: The is the URL to the location where PPS Azure vhd image is stored.

```
"PPSImageLocationURI": {
  "type": "string",
   "defaultValue": "https://polsecstorageaccount.blob.core.windows.net/90r3images/PSA-V-HYPERV-PPS-51773.1-SERIAL-hyperv.vhd",
  "metadata": {
    "description": "URL of Pulse Policy Secure vhd image"
  }
},
```

PPS VM Name: This is the name given to PPS Virtual Appliance.

```
"PPSVMName": {
  "type": "string",
  "defaultValue": "PPSAzureVA",
  "metadata": {
    "description": "Pulse Policy Secure VA Name"
   }
},
```

SSH Public Key: This is an RSA public key that is used to access Pulse Policy Secure via SSH.

```
"SSHPublicKey": {
  "type": "string",
  "metadata": {
    "description": "Provide an RSA public key. This key is used to access PPS via SSH. You can generate SSH keys using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. SSH Publ
  }
},
```

PPS Config: This section contains provisioning parameters that are required during the deployment of a Virtual Appliance. An XML-based configuration file can be present in another Virtual Machine in Azure cloud or in the corporate network which is accessible for Pulse Policy Secure through site-to-site VPN between Azure and the corporate data center.

Pulse Policy Secure accepts the following parameters as provisioning parameters:

- primary-dns
- secondary-dns
- wins-server

- dns-domain
- username
- ssh-publickey
- cert-common-name
- cert-random-text
- cert-organization
- config-download-url
- config-data
- auth-code-license
- enable-license-server
- accept-license-agreement
- enable-rest

For details about these parameters, see Pulse Policy Secure Provisioning Parameters.

```
"<pulse-config><primary-dns>8.8.8.8</primary-dns><secondary-dns>8.8.8.9</secondary-dns><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net</dns-dom
ain><admin-username>admin</admin-username><admin-password>password</admin-password><cert-common-name>va1.psecure.net</cert-common-name><cert-random-text>
fdsfpisonvsfnms</cert-random-text><cert-organisation>Psecure
Org</cert-organisation><config-download-url></config-download-url><config-data></config-data><auth-code-license></auth-code-license><enable-license-serve
r>n</enable-license-server><accept-license-agreement>n</accept-license-agreement><enable-rest>n</enable-rest></pulse-config>",
```

DNS Label Prefix Mgmt: This is the prefix for Management Interface DNS label.

```
"dnsLabelPrefixMgmt": {
  "type": "string",
 "defaultValue": "mycloudppsmgmt",
  "metadata": {
    "description": "Unique DNS Name for the Public IP used to access PPS"
  }
},
```

VNet Address Space: This is a Virtual Network address space.

```
"VnetAddressSpace": {
    "type": "string",
    "defaultValue": "10.100.0.0/16",
    "metadata": {
        "description": "Virtual Network Address Space"
    }
},
```

Internal Subnet: Subnet from which Pulse Policy Secure Internal Interface needs to lease IP.

```
"InternalSubnet": {
    "type": "string",
    "defaultValue": "10.100.1.0/24",
    "metadata": {
        "description": "PPS internal interface connects to this subnet"
    }
},
```

External Subnet: Subnet from which Pulse Policy Secure External Interface needs to lease IP.

```
"ExternalSubnet": {
    "type": "string",
    "defaultValue": "10.100.2.0/24",
    "metadata": {
        "description": "PPS external interface connects to this subnet"
    }
},
```

Management Subnet: Subnet from which Pulse Policy Secure Management Interface needs to lease IP.

```
"ManagementSubnet": {
    "type": "string",
    "defaultValue": "10.100.3.0/24",
    "metadata": {
        "description": "PPS management interface connects to this subnet"
    }
}
```

variables

PPS Virtual Network: This is the variable associated with the PPS Virtual Network.

```
"ppsvnetname"                : "PPSVirtualNetwork",
```

PPS Internal Subnet: This is the variable associated with the Subnet from which Pulse Policy Secure Internal Interface needs to lease IP.

```
"ppsVnetIntSubnet"           : "PPSInternalSubnet",
```

PPS External Subnet: This is the variable associated with the Subnet from which Pulse Policy Secure External Interface needs to lease IP

```
"ppsVnetExtSubnet"           : "PPSExternalSubnet",
```

PPS Management Subnet: This is the variable associated with the Subnet from which Pulse Policy Secure Management Interface needs to lease IP.

```
"ppsVnetMgmtSubnet"          : "PPSManagemnetSubnet",
```

CS Internal Private IP: This is the private IP address of the Internal IP.

```
"ppsIntPrivateIP"            : "10.100.1.4",
```

PPS Internal NIC: This is network interface card of PPS Internal network.

```
"ppsIntNic"                  : "PPSInternalNIC",
```

PPS External NIC: This is network interface card of PPS External network.

```
"ppsExtNic"                  : "PPSExternalNIC",
```

PPS Management NIC: This is network interface card of PPS Management network.

```
"ppsMgmtNic"                 : "PPSMangementNIC",
```

PPS Management Public IP: This is public IP address assigned to PPS Management Subnet.

```
"publicIPAddr1"              : "PPSManagementPublicIP",
```

Public IP Address Type: This variable is defined as static IP.

```
"publicIPAddressType"        : "Static",
```

NSG Internal Subnet: This variable defines Network Security Group's Internal Subnet policy.

```
"nsgInt"                     : "NSGInternalSubnet",
```

NSG External Subnet: This variable defines Network Security Group's External Subnet policy.

```
"nsgExt"                     : "NSGExternalSubnet",
```

NSG Management Subnet: This variable defines Network Security Group's Management Subnet policy.

```
"nsgMgmt"                    : "NSGManagementSubnet",
```

VM Size: This variable defines PPS Virtual Machine size. It is 4 cores, 144MB memory.

```
"vmSize"                     : "Standard_DS3_v2",
```

Virtual Network ID: This variable defines PPS Virtual Network name.

```
"vnetID"                     : "[resourceId('Microsoft.Network/virtualNetworks',variables('pcsvnetname'))]",

"subnetRefInt"               : "[concat(variables('vnetID'),'/subnets/',variables('pcsVnetIntSubnet'))]",

"subnetRefExt"               : "[concat(variables('vnetID'),'/subnets/',variables('pcsVnetExtSubnet'))]",

"subnetRefMgmt"              : "[concat(variables('vnetID'),'/subnets/',variables('pcsVnetMgmtSubnet'))]",
```

API Version

```
"apiVersion"                 : "2015-06-15"
```

resources

publicIPAddresses/publicIPAddr1: This block is responsible for creating public IP address which is static in nature. This is used for management interface IP address of PPS.

```
"type": "Microsoft.Network/publicIPAddresses",
"name": "[variables('publicIPAddr1')]",
```

virtualNetworks/ppsvnetname: This block is responsible for creating PPS Virtual Network name. The creation of PPS Virtual Network name depends on:

- NSG Internal Subnet
- NSG External Subnet
- NSG Management Subnet

```
"type": "Microsoft.Network/virtualNetworks",
"name": "[variables('ppsvnetname')]",
```

virtualNetworks/ppsVnetIntSubnet: This block is responsible for creating subnet. The created subnet is applied to PPS Internal interface.

```
"name": "[variables('ppsVnetIntSubnet')]",
```

virtualNetworks/ppsVnetExtSubnet: This block is responsible for creating subnet. The created subnet is applied to PPS External interface.

```
"name": "[variables('ppsVnetExtSubnet')]",
```

virtualNetworks/ppsVnetMgmtSubnet: This block is responsible for creating subnet. The created subnet is applied to PPS Management interface.

```
"name": "[variables('ppsVnetMgmtSubnet')]",
```

networkInterfaces/ppsExtNic: This block is responsible for creating network interface. The created network interface is applied to network interface card of PPS External interface. The creation of this network interface depends on:

- PPS Virtual Network name
- Public IP address of External Subnet

```
"type": "Microsoft.Network/networkInterfaces",
"name": "[variables('ppsExtNic')]",
```

networkInterfaces/ppsMgmtNic: This block is responsible for creating network interface. The created network interface is applied to network interface card of PPS Management interface. The creation of this network interface depends on:

- PPS Virtual Network name
- Public IP address of Management Subnet

```
"type": "Microsoft.Network/networkInterfaces",
"name": "[variables('ppsMgmtNic')]",
```

networkInterfaces/ppsIntNic: This block is responsible for creating network interface. The created network interface is applied to network interface card of PPS Internal interface. The creation of this network interface depends on:

- PPS Virtual Network name

```
"type": "Microsoft.Network/networkInterfaces",
"name": "[variables('ppsIntNic')]",
```

virtualMachines/PPSVmName: This block is responsible for creating Virtual Machine name. The created Virtual machine name is applied to PPS Virtual Machine. The creation of PPS Virtual Machine name depends on:

- Network Interface Card of PPS Internal interface
- Network Interface Card of PPS External interface
- Network Interface Card of PPS Management interface

```
"type": "Microsoft.Compute/virtualMachines",
"name": "[parameters('PPSVmName')]",
```

networkSecurityGroups/nsgExt: This block is responsible for creating policy. The created policy is applied to Network Security Group's External interface.

```
"type": "Microsoft.Network/networkSecurityGroups",
"name": "[variables('nsgExt')]",
```

The following security rules can be defined:

- allowHTTPS
- allowHTTP
- allowRADIUSAuth
- allowRADIUSAccounting
- allow RADIUSCoA
- allowSyslogUDP
- allowSyslogTCP
- allowSNMPTrap
- allowSRXConnection
- allowScreenOSConn
- allowDHCP
- allowTACACSPlus
- allowPostgresDataSync

networkSecurityGroups/nsgMgmt: This block is responsible for creating policy. The created policy is applied to Network Security Group's Management interface.

```
"type": "Microsoft.Network/networkSecurityGroups",
"name": "[variables('nsgMgmt')]",
```

The following security rules can be defined:

- allowHTTPS
- allowHTTP
- allowCustomSSH
- allowDMI
- denyAll

networkSecurityGroups/nsgInt: This block is responsible for creating policy. The created policy is applied to Network Security Group's Internal interface.

```
"type": "Microsoft.Network/networkSecurityGroups",
"name": "[variables('nsgInt')]",
```

The following security rules can be defined:

- allowHTTPS
- allowHTTP
- allowRADIUSAuth
- allowRADIUSAccounting
- allowRADIUSCoA
- allowSyslogUDP
- allowSyslogTCP
- allowSNMPTrap
- allowSRXConnection
- allowScreenOSConn
- allowDHCP
- allowTACACSPlus
- allowPostgresDataSync
- allowCustomSSH

outputs

The outputs section defines the public IP address and FQDN that is displayed on successful deployment of PPS on Azure.

```
"outputs": {
  "hostname": {
    "type": "string",
    "value": "[reference(variables('publicIPAddr1')).dnsSettings.fqdn]"
  }
```

Appendix C: Pulse Policy Secure Resource Manager Template for an Existing Virtual Network

Pulse Secure provides sample Azure template files to deploy Pulse Policy Secure Virtual Appliance on Azure. Users can modify this to make it suitable for their need. Visit https://www.pulsesecure.net and download the pulsesecure-pps-3-nics.zip file, and unzip it to get **azuredeploy.json**.

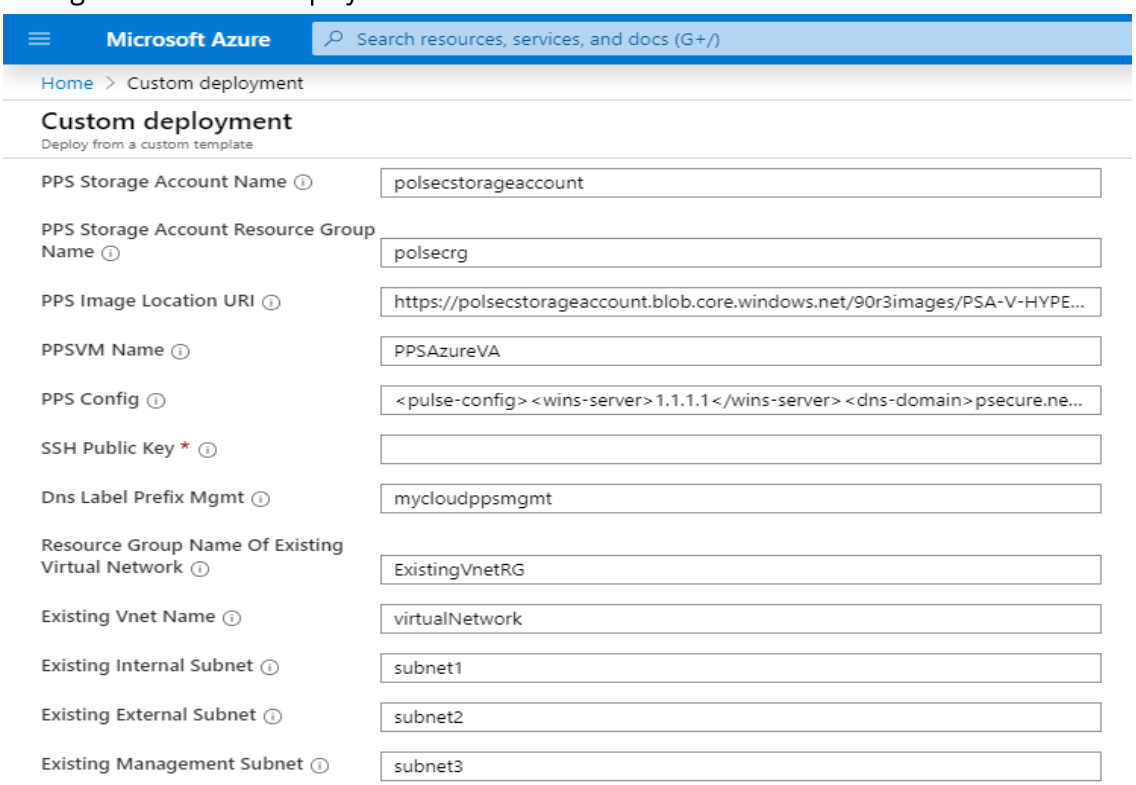
This template creates a new PPS with 3 NICs, Vnet, three subnets, NSG policies attached to PPS internal, external and management subnets. All 3 NICs of PPS are configured with dynamic IP configuration and enabled IP forwarding. Public IPs are attached to the PPS external and management NIC.

The template has following sections:

parameters	This section defines the parameters used for deploying PPS on Azure. It contains parameter name, its default value and the mouse-over help text that is displayed when mouse is placed over the parameter in Azure Web portal. The parameters defined here are displayed in the Custom Deployment page of Azure portal.
variables	This section defines variables that will be used in the functions defined in the resources section.
resources	This section defines resource types that are deployed or updated in a resource group.
outputs	This section defines the public IP address and FQDN returned after successful deployment of PPS on Azure.

parameters

Figure 52: Custom Deployment



PPS Storage Account Name: This is the name of the PPS Storage Account where the PPS Azure vhd image is stored.

```
"parameters": {
    "PPSStorageAccountName": {
    "type": "string",
    "defaultValue": "polsecstorageaccount",
    "metadata": {
      "description": "Storage account name where PPS image is uploaded"
    }
  },
```

PPS Storage Account Resource Group Name: The is the name of the PPS Storage Account Resource Group where the PPS Azure vhd image is stored.

```
"PPSStorageAccountResourceGroupName": {
  "type": "string",
  "defaultValue": "polsecrg",
  "metadata": {
    "description": "Resource group of the existing storage account where PPS image is uploaded"
  }
},
```

PPS Image Location URI: The is the URL to the location where PPS Azure vhd image is stored.

```
"PPSImageLocationURI": {
  "type": "string",
   "defaultValue": "https://polsecstorageaccount.blob.core.windows.net/90r3images/PSA-V-HYPERV-PPS-51773.1-SERIAL-hyperv.vhd",
  "metadata": {
    "description": "URL of Pulse Policy Secure vhd image"
  }
},
```

PPS VM Name: This is the name given to Pulse Policy Secure Virtual Appliance.

```
"PPSVMName": {
  "type": "string",
  "defaultValue": "PPSAzureVA",
  "metadata": {
    "description": "Pulse Policy Secure VA Name"
   }
},
```

PPS Config: This section contains provisioning parameters that are required during the deployment of a Virtual Appliance. An XML-based configuration file can be present in another Virtual Machine in Azure cloud or in the corporate network which is accessible for Pulse Policy Secure through site-to-site VPN between Azure and the corporate data center.
Pulse Policy Secure accepts the following parameters as provisioning parameters:

- primary-dns
- secondary-dns
- wins-server
- dns-domain
- username
- ssh-publickey
- cert-common-name
- cert-random-text
- cert-organization
- config-download-url
- config-data
- auth-code-license
- enable-license-server
- accept-license-agreement
- enable-rest

For details about these parameters, see Pulse Policy Secure Provisioning Parameters.


```
"<pulse-config><primary-dns>8.8.8.8</primary-dns><secondary-dns>8.8.8.9</secondary-dns><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net</dns-dom
ain><admin-username>admin</admin-username><admin-password>password</admin-password><cert-common-name>va1.psecure.net</cert-common-name><cert-random-text>
fdsfpisonvsfnms</cert-random-text><cert-organisation>Psecure
Org</cert-organisation><config-download-url></config-download-url><config-data></config-data><auth-code-license></auth-code-license><enable-license-serve
r>n</enable-license-server><accept-license-agreement>n</accept-license-agreement><enable-rest>n</enable-rest></pulse-config>",
```


SSH Public Key: This is an RSA public key that is used to access Pulse Policy Secure via SSH.


```
"SSHPublicKey": {
  "type": "string",
  "metadata": {
    "description": "Provide an RSA public key. This key is used to access PPS via SSH. You can generate SSH keys using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. SSH Publ
  }
},
```


DNS Label Prefix Mgmt: This is the prefix for Management Interface DNS label.


```
  "dnsLabelPrefixMgmt": {
    "type": "string",
  "defaultValue": "mycloudppsmgmt",
    "metadata": {
      "description": "Unique DNS Name for the Public IP used to access PPS"
    }
  },
```


Resource Group Name of Exiting Virtual Network: Name of the Resource Group that contains the existing Virtual network.


```
"ResourceGroupNameOfExistingVirtualNetwork": {
    "type": "string",
    "defaultValue": "ExistingVnetRG",
    "metadata": {
        "description": "Name of the resource group that contains the existing virutal network."
    }
},
```


Existing Virtual Network Name: Name of the existing Virtual network.


```
 "existingVnetName": {
    "type": "string",
    "defaultValue": "virtualNetwork",
    "metadata": {
        "description": "Name of existing virtual network"
    }
 },
```


Existing Internal Subnet: Subnet from which Pulse Policy Secure Internal Interface needs to lease IP.


```
"existingInternalSubnet": {
    "type": "string",
    "defaultValue": "subnet1",
    "metadata": {
        "description": "PPS internal interface connects to this subnet"
    }
},
```


Existing External Subnet: Subnet from which Pulse Policy Secure External Interface needs to lease IP.


```
"existingExternalSubnet": {
    "type": "string",
    "defaultValue": "subnet2",
    "metadata": {
        "description": "PPS external interface connects to this subnet"
    }
},
```


Existing Management Subnet: Subnet from which Pulse Policy Secure Management Interface needs to lease IP.

```
"existingManagementSubnet": {
    "type": "string",
    "defaultValue": "subnet3",
    "metadata": {
        "description": "PPS management interface connects to this subnet"
    }
}
```

variables

PPS Internal NIC: This is network interface card of PPS Internal network.

```
"ppsIntNic"                    : "PPSInternalNIC",
```

PPS External NIC: This is network interface card of PPS External network.

```
"ppsExtNic"                    : "PPSExternalNIC",
```

PPS Management NIC: This is network interface card of PPS Management network.

```
"ppsMgmtNic"                   : "PPSMangementNIC",
```

PPS Management Public IP: This is public IP address assigned to PPS Management Subnet.

```
"publicIPAddr1"                : "PPSManagementPublicIP",
```

Public IP Address Type: This variable is defined as static IP.

```
"publicIPAddressType"          : "Static",
```

NSG Internal Subnet: This variable defines Network Security Group's Internal Subnet policy.

```
"nsgInt"                       : "NSGInternalSubnet",
```

NSG External Subnet: This variable defines Network Security Group's External Subnet policy.

```
"nsgExt"                       : "NSGExternalSubnet",
```

NSG Management Subnet: This variable defines Network Security Group's Management Subnet policy.

```
"nsgMgmt"                      : "NSGManagementSubnet",
```

VM Size: This variable defines PPS Virtual Machine size. It is 4 cores, 144MB memory.

```
"vmSize"                       : "Standard_DS3_v2",
```

Virtual Network ID: This variable defines PPS Virtual Network name.

```
"vnetID"                       : "[resourceId('Microsoft.Network/virtualNetworks',variables('pcsvnetname'))]",

"subnetRefInt"                 : "[concat(variables('vnetID'),'/subnets/',variables('pcsVnetIntSubnet'))]",

"subnetRefExt"                 : "[concat(variables('vnetID'),'/subnets/',variables('pcsVnetExtSubnet'))]",

"subnetRefMgmt"                : "[concat(variables('vnetID'),'/subnets/',variables('pcsVnetMgmtSubnet'))]",

"subnetRefTunnel"              : "[concat(variables('vnetID'),'/subnets/',parameters('existingTunnelSubnet'))]",
```

API Version

```
"apiVersion"                    : "2015-06-15"
```

resources

publicIPAddresses/publicIPAddr1: This block is responsible for creating public IP address which is static in nature. This is used for management interface IP address of PPS.

```
"type": "Microsoft.Network/publicIPAddresses",
"name": "[variables('publicIPAddr1')]",
```

networkSecurityGroups/nsgExt: This block is responsible for creating policy. The created policy is applied to Network Security Group's External interface.

```
"type": "Microsoft.Network/networkSecurityGroups",
"name": "[variables('nsgExt')]",
```

The following security rules can be defined:

- allowHTTPS
- allowHTTP
- allowRADIUSAuth
- allowRADIUSAccounting
- allowRADIUSCoA
- allowSyslogUDP
- allowSyslogTCP
- allowSNMPTrap
- allowSRXConnection
- allowScreenOSConn
- allowDHCP
- allowTACACSPlus
- allowPostgresDataSync

networkSecurityGroups/nsgMgmt: This block is responsible for creating policy. The created policy is applied to Network Security Group's Management interface.

```
"type": "Microsoft.Network/networkSecurityGroups",
"name": "[variables('nsgMgmt')]",
```

The following security rules can be defined:

- allowHTTPS
- allowHTTP
- allowCustomSSH
- allowDMI
- denyAll

networkSecurityGroups/nsgInt: This block is responsible for creating policy. The created policy is applied to Network Security Group's Internal interface.

```
"type": "Microsoft.Network/networkSecurityGroups",
"name": "[variables('nsgInt')]",
```

The following security rules can be defined:

- allowHTTPS

- allowHTTP
- allowRADIUSAuth
- allowRADIUSAccounting
- allow RADIUSCoA
- allowSyslogUDP
- allowSyslogTCP
- allowSNMPTrap
- allowSRXConnection
- allowScreenOSConn
- allowDHCP
- allowTACACSPlus
- allowPostgresDataSync
- allowCustomSSH

networkInterfaces/ppsExtNic: This block is responsible for creating network interface. The created network interface is applied to network interface card of PPS External interface. The creation of this network interface depends on:

- PPS Virtual Network name
- Public IP address of External Subnet

```
"type": "Microsoft.Network/networkInterfaces",
"name": "[variables('ppsExtNic')]",
```

networkInterfaces/ppsMgmtNic: This block is responsible for creating network interface. The created network interface is applied to network interface card of PPS Management interface. The creation of this network interface depends on:

- PPS Virtual Network name
- Public IP address of Management Subnet

```
"type": "Microsoft.Network/networkInterfaces",
"name": "[variables('ppsMgmtNic')]",
```

networkInterfaces/ppsIntNic: This block is responsible for creating network interface. The created network interface is applied to network interface card of PPS Internal interface. The creation of this network interface depends on:

- PPS Virtual Network name

```
"type": "Microsoft.Network/networkInterfaces",
"name": "[variables('ppsIntNic')]",
```

virtualMachines/PPSVmName: This block is responsible for creating Virtual Machine name. The created Virtual machine name is applied to PPS Virtual Machine. The creation of PPS Virtual Machine name depends on:

- Network Interface Card of PPS Internal interface
- Network Interface Card of PPS External interface
- Network Interface Card of PPS Management interface

```
"type": "Microsoft.Compute/virtualMachines",
"name": "[parameters('PPSVmName')]",
```

outputs

The outputs section defines the public IP address and FQDN that is displayed on successful deployment of PPS on Azure.

```
"outputs": {
  "hostname": {
    "type": "string",
    "value": "[reference(variables('publicIPAddr1')).dnsSettings.fqdn]"
  }
```

References

Microsoft Azure documentation: https://docs.microsoft.com/en-us/azure/

Requesting Technical Support

Technical product support is available through the Pulse Secure Global Support Center (PSGSC). If you have a support contract, then file a ticket with PSGSC.

- Product warranties—for product warranty information, visit https://www.pulsesecure.net.