Pulse Secure®

Pulse Connect Secure Virtual Appliance on Microsoft Azure

Deployment Guide

Document Revision	3.1
Published Date	October 2019

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

https://www.pulsesecure.net

Pulse Secure and the Pulse Secure logo are trademarks of Pulse Secure, LLC in the United States. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Pulse Secure, LLC assumes no responsibility for any inaccuracies in this document. Pulse Secure, LLC reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Pulse Secure or components thereof might be covered by one or more of the following patents that are owned by or licensed to Pulse Secure: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Pulse Connect 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

The Pulse Secure product that is the subject of this technical documentation consists of (or is intended for use with) Pulse Secure software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.pulsesecure.net/product-service-policies/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

© 2019 Pulse Secure, LLC. All rights reserved

Revision History

Revision and Date	Added/Updated /Removed	Remarks
3.1 October 2019	None	Cosmetic changes
3.0 July 2019	Added the "Backing up Configs and Archived Logs on Azure Storage" section. Limitations section is updated.	
2.0 April 2019	Updated FAQ section with Source NATTING feature.	
1.2 March 2019	Updated the "Overview", "Deploying PCS Active-Active Cluster using Virtual Traffic Manager in Microsoft Azure" and "Limitations" sections.	
1.1 December 2018	Added "Deploying PCS Active-Active Cluster using Virtual Traffic Manager in Microsoft Azure" section.	
1.0, September 2018	None	No changes to the document from the previous release.

© 2019 Pulse Secure, LLC. All rights reserved

Table of Contents

Revision History 3
Overview 6
About This Guide 6
Assumptions 6
Pulse Connect Secure on Azure Marketplace 7
Prerequisites and System Requirements on Azure Marketplace 7
Deploying Pulse Connect Secure on Azure Marketplace 7
Basic Configuration 8
Network Settings 9
Instance Configuration 10
Summary Step 11
Pulse Connect Secure on Microsoft Azure Cloud 12
Prerequisites and System Requirements on Azure 13
Steps to Deploy Pulse Connect Secure on Azure 13
Upload Pulse Connect Secure Virtual Appliance Image to Azure Web Portal 13
Upload Azure Resource Manager Template to Azure Account 15
Deploying Pulse Connect Secure on Azure using Azure Portal 17
Deploying PCS on New Virtual Network 17
Deployment on VM with Three NIC Cards 17
Deployment on VM with Two NIC Cards 21
Deploying PCS on an Existing Virtual Network 24
Deployment on VM with Three NIC Cards 24
Deployment on VM with Two NIC Cards 28
Deploying Pulse Connect Secure on Azure using Azure CLI 31
Pulse Connect Secure Provisioning Parameters 33
Provisioning Pulse Connect Secure with Predefined Configuration 34
Configuring Licenses on the Pulse Connect Secure Appliance 35
Pulse License Server in Corporate Network 35
Pulse License Server in Cloud Network 35
Adding Authentication Code in PCS Admin Console 36
Including Authentication Code in ARM Template 36
Deploying PCS Active-Active Cluster using Virtual Traffic Manager in Microsoft Azure 37
Deploying Two PCS EC2 instances Using ARM Template 37
Forming the Active-Active Cluster 37
Deploying Virtual Traffic Manager EC2 Instance in the External Subnet of PCS in Microsoft Azure 38
Setting Up and Configuring vTM for External Users 44
Accessing the Pulse Connect Secure Virtual Appliance 48
Accessing the Pulse Connect Secure Virtual Appliance as an Administrator 48
Accessing the Pulse Connect Secure Virtual Appliance as an End User 49
Accessing the Pulse Connect Secure Virtual Appliance using SSH Console 49
On Linux and Mac OSX 49

© 2019 Pulse Secure, LLC. All rights reserved

On Windows 49
System Operations 51
Network Configuration 51
IP Address Assignment for Internal, External and Management Interfaces 51
IP Addressing Modes 51
Modifying Network Parameters After Deployment 52
Controlling the Selection of Internal, External and Management Interfaces 52
Backing up Configs and Archived Logs on Azure Storage 54
Configuring Backup Configs and Archived Logs via PCS Admin Console 54
Configuring Backup Configs and Archived Logs via REST 55
Setting Azure as Archive Logs Backup 55
Decommissioning Pulse Connect Secure 55
Delete Entire Resource Group that the Pulse Connect Secure Is In 55
Delete Pulse Connect Secure and Resource It Uses, but not the Other Resources in Resource Group 56
Pricing 57
Limitations 57
Not Qualified 57
Troubleshooting 58
Frequently Asked Questions 59
FAQ1: I am unable to connect to my backend resources through L3 VPN 59
Testing the Connection to CentOS System 61
Testing the Connection to On-premise Resource 64
FAQ2: Users are unable to access internet resources when connected to a VPN tunnel on an Azure-based PCS 65
Appendix A: Network Security Group (NSG) 66
Appendix B: Pulse Connect Secure Resource Manager Template 70
parameters 70
variables 73
resources 74
outputs 77
Appendix C: Pulse Connect Secure Resource Manager Template for an Existing Virtual Network 78
parameters 78
variables 81
resources 82
outputs 84
References 85
Requesting Technical Support 85

© 2019 Pulse Secure, LLC. All rights reserved

Overview

About This Guide

This guide helps in deploying the Pulse Connect Secure Virtual Appliance on Microsoft Azure. Beginning 9.0R3 release, Pulse Connect Secure is made available in Azure Marketplace. The PCS 9.1R1 image is now available in Azure Marketplace.
This document also describes how a Pulse Connect Secure administrator manually upload the Pulse Connect Secure Virtual Appliance image into Microsoft Azure storage account. And, once the image is available in the Azure storage account, how the Pulse Connect Secure administrator can deploy Pulse Connect Secure on Microsoft Azure.

Assumptions

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

© 2019 Pulse Secure, LLC. All rights reserved

Pulse Connect Secure on Azure Marketplace

Prerequisites and System Requirements on Azure Marketplace

To deploy the Pulse Connect 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 Connect Secure licenses *

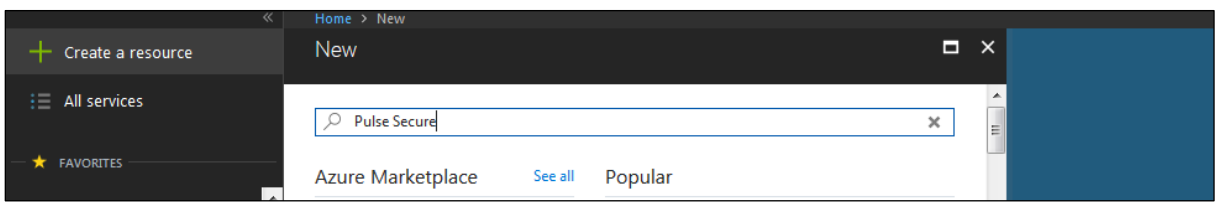
Note:
* Pulse Connect Secure Virtual Appliance, by default, has two-users license. This release supports licensing with License server located at corporate network and licensing through Pulse Cloud Licensing Service (PCLS) server. For licensing through PCLS, administrator needs to obtain Authentication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

Note: From release 9.0R1 onwards, PCS supports VM with 2-NICs model and 3-NICs model for deployment.

Deploying Pulse Connect Secure on Azure Marketplace

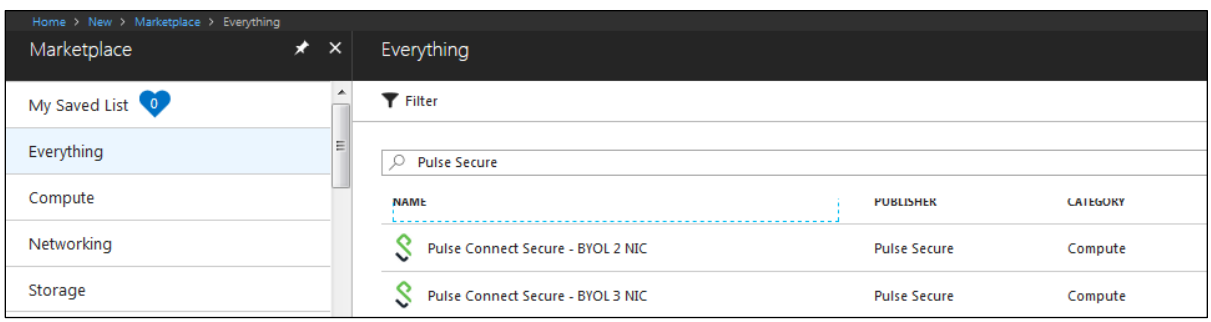
1. Log into Azure portal and navigate to Azure Marketplace by clicking **Create a resource**.

   Figure 1: Marketplace



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

   Figure 2: Published Pulse Secure Images



Azure Marketplace contains the following two Pulse Connect Secure SKUs:

- Pulse Connect Secure-BYOL 2 NIC
- Pulse Connect Secure BYOL 3 NIC

© 2019 Pulse Secure, LLC. All rights reserved

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

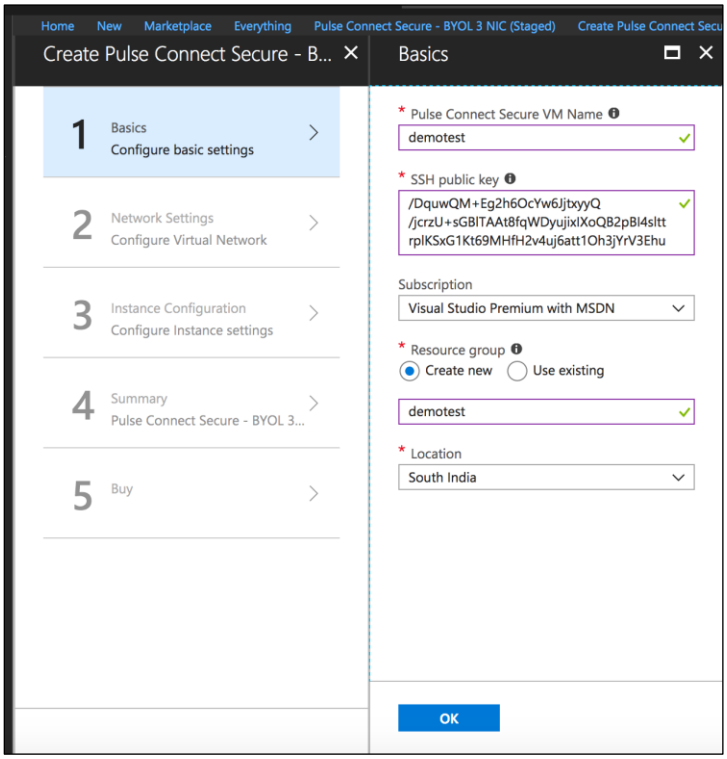
Basic Configuration

4. In the Basic Configuration step, enter the following parameters and click **OK**:

- **VM name**: Name of the Pulse Connect 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 PCS 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**: Name of the resource group to be deployed.

Figure 3: Basic Configuration Settings



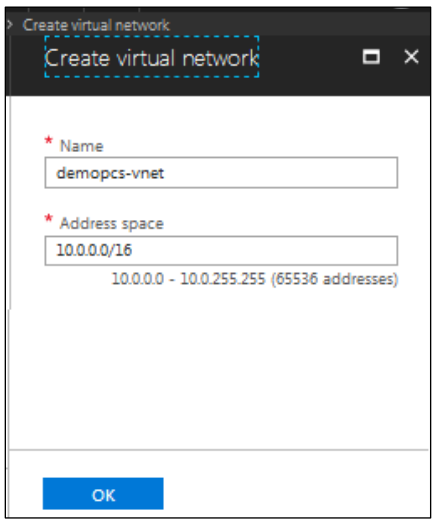
© 2019 Pulse Secure, LLC. All rights reserved

Network Settings

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

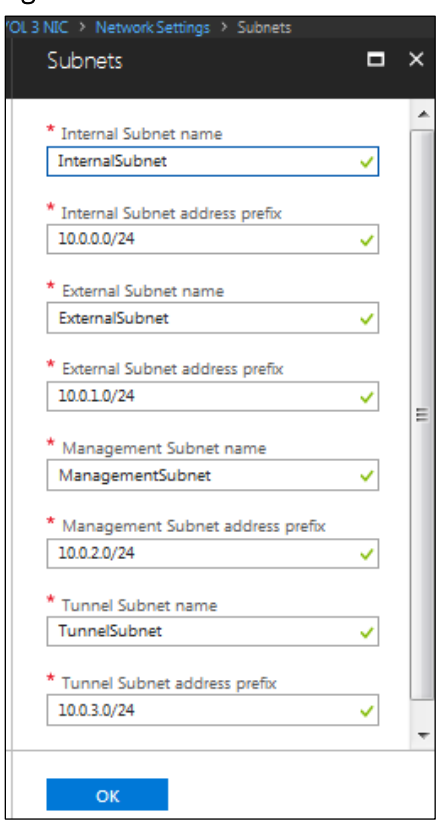
- Virtual Network:
  - Select an existing virtual network from the list or
  - Create a new virtual network. Specify the virtual network name and the address space.

Figure 4: Virtual Network



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

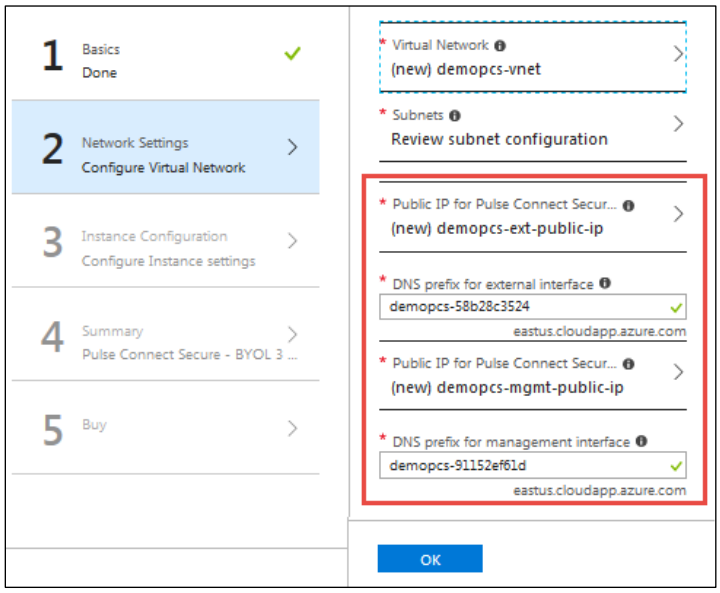
Figure 5: Subnets



© 2019 Pulse Secure, LLC. All rights reserved

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

Figure 6: Network Settings



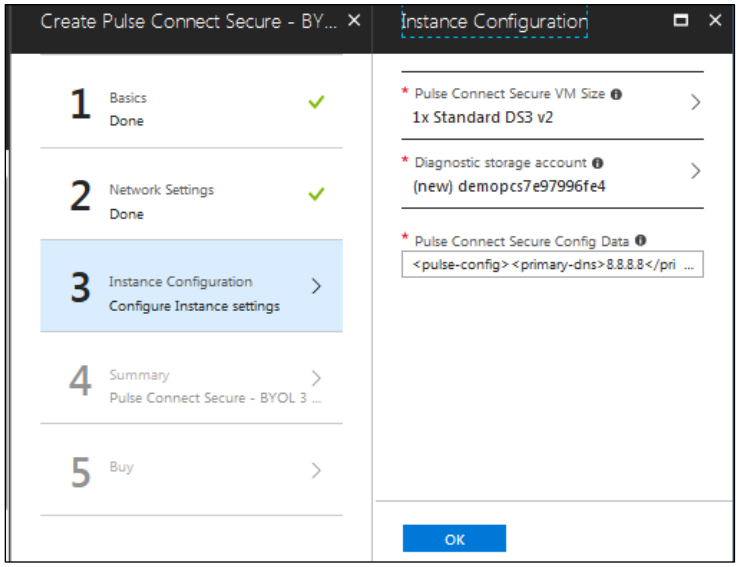
Instance Configuration

6. In the Instance Configuration step, enter the following parameters:

- **Pulse Connect 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.
- **Diagnostic storage account**: Storage account for the Virtual Machine's diagnostics
- **Pulse Connect Secure Config Data**: Provisioning parameters in an XML format. Refer the section "Pulse Connect Secure Provisioning Parameters'

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

Figure 7: Instance Configuration

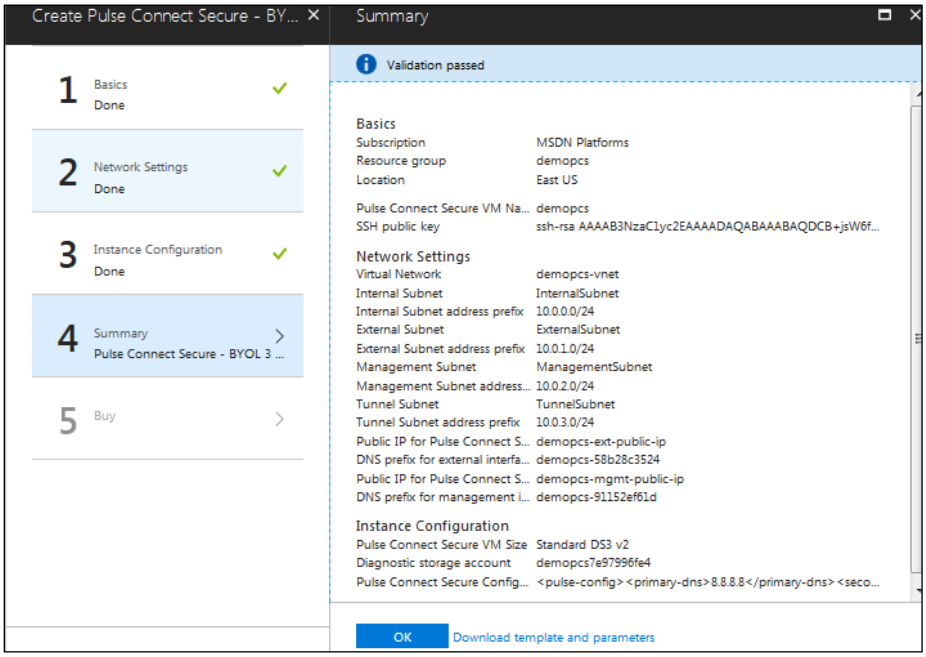


© 2019 Pulse Secure, LLC. All rights reserved

Summary Step

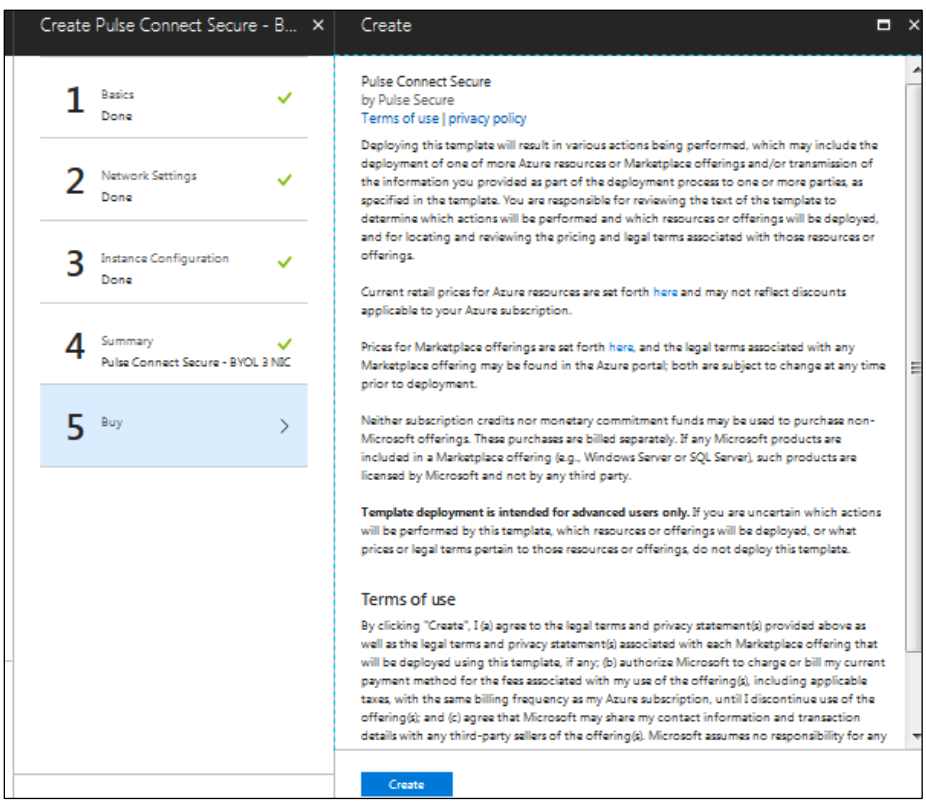
7. In the Summary step, once the final validation is complete, click **OK**.

Figure 8: Configuration Validation



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

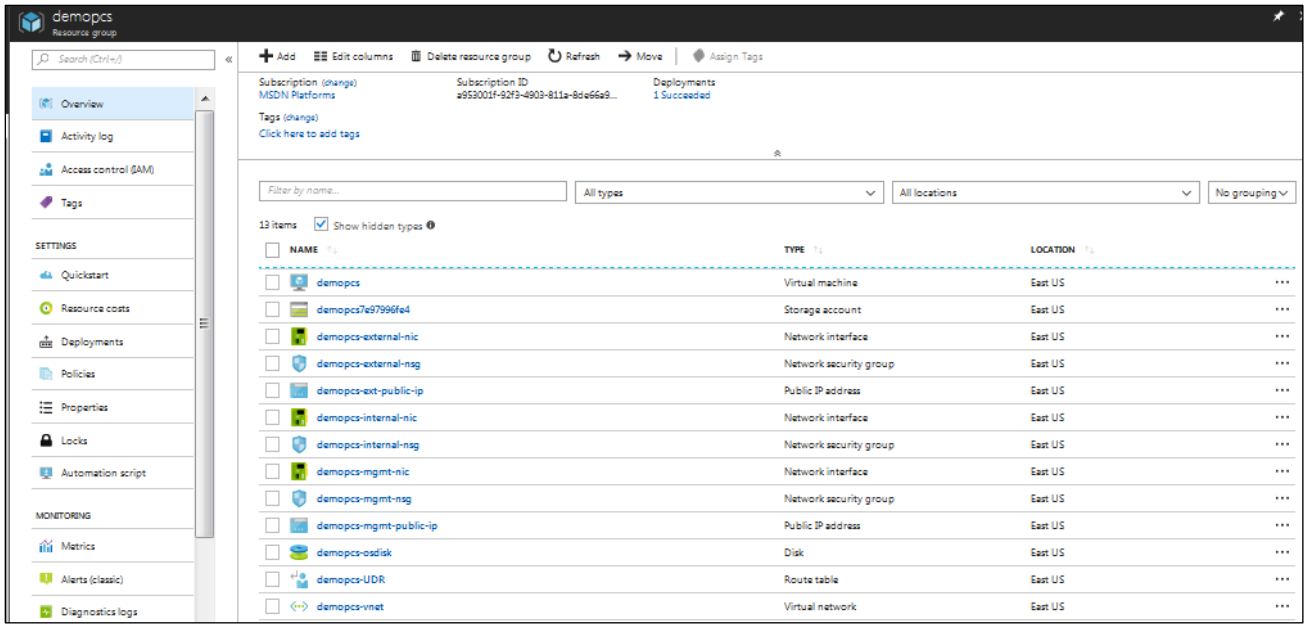
Figure 9: Terms of Use



© 2019 Pulse Secure, LLC. All rights reserved

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

Figure 10: Deployment in Progress



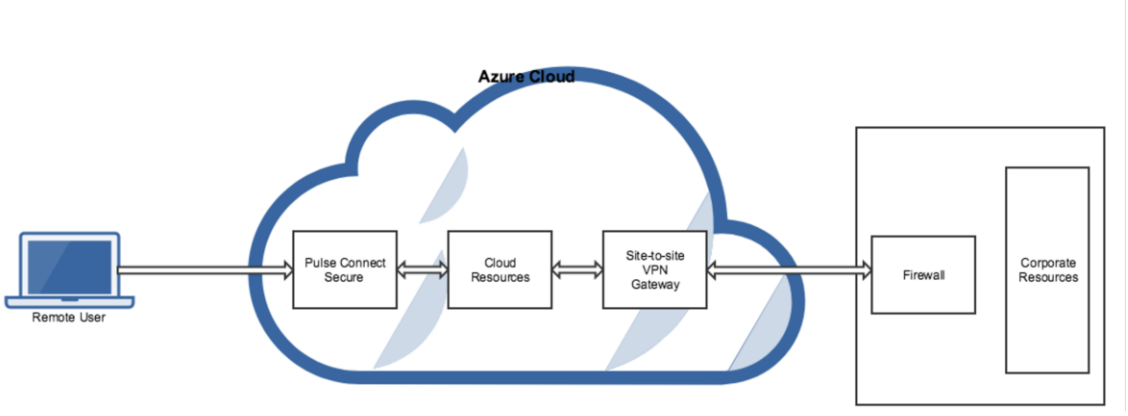
Note: For L3 connectivity, make sure that "<PCS VM name>-UDR" is properly associated with the subnet where Backend/Protected resources are connected. For example:

- If PCS internal, external and management interfaces are connected to subnet1, subnet2 and subnet3 respectively and Backend/Protected resources are in subnet5, then we need to associate <PCS VM name>-UDR to subnet5.
- If Backend/Protected resources are in Datacenter or in different virtual network, then associate <PCS VM name>-UDR to GatewaySubnet.

Pulse Connect Secure on Microsoft Azure Cloud

As depicted in the below diagram, a remote user can use Pulse Connect Secure to securely access cloud resources as well as corporate resources. To access corporate resources, the Pulse Connect Secure administrator needs to ensure that site-to-site VPN is already established between Azure and corporate network.

Figure 11: Pulse Connect Secure on Microsoft Azure



© 2019 Pulse Secure, LLC. All rights reserved

Prerequisites and System Requirements on Azure

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

- A Microsoft Azure account
- Access to the Microsoft Azure portal (https://portal.azure.com)*
- Pulse Connect Secure Virtual Appliance Image (.vhd file)
- Azure Resource Manager template (ARM template)
- Pulse Connect Secure licenses **
- Site-to-Site VPN between Azure and the corporate network (optional)

  Note: This is needed only if the Pulse Connect Secure users need to access corporate resources.
- Pulse License Server (optional)**
  - Located at corporate network, accessible through site-to-site VPN
- Pulse Connect Secure configuration in XML format (optional)
- The following systems are qualified in 9.0R1 release:
  - DS2 – 2-core
  - DS3 – 4-core
  - DS4 – 8-core

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

Note: From release 9.0R1 onwards, PCS supports VM with 2-NICs model and 3-NICs model for deployment.

Steps to Deploy Pulse Connect Secure on Azure

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

- Upload Pulse Connect 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 Connect Secure.

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

Upload Pulse Connect Secure Virtual Appliance Image to Azure Web Portal

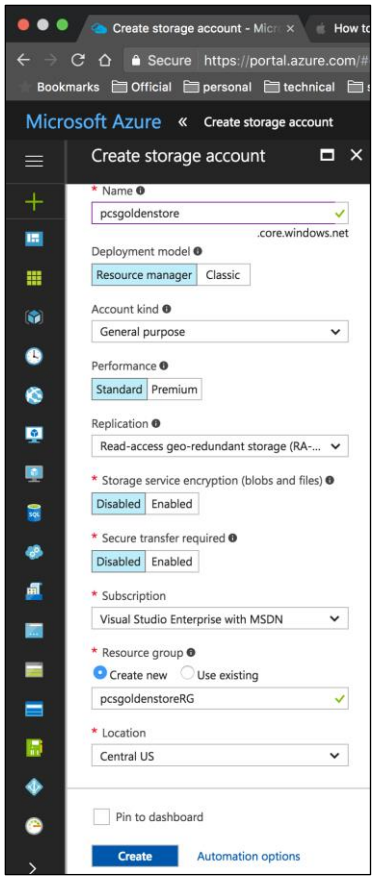
This section shows the steps to upload the Pulse Connect Secure Virtual Appliance image to Azure web portal.
To upload Pulse Connect Secure Virtual Appliance image to Azure web portal, do the following:

1. Visit the Pulse Secure support site www.pulsesecure.net and download the Azure PCS image file (**ps-pcs-azure-psa-v-<releaseno>-<buildno>-package.zip**) which is in the zipped format.
2. Unzip the file and look for the Pulse Connect Secure Virtual Appliance **vhd** image.
3. Log in to the Azure portal.

© 2019 Pulse Secure, LLC. All rights reserved

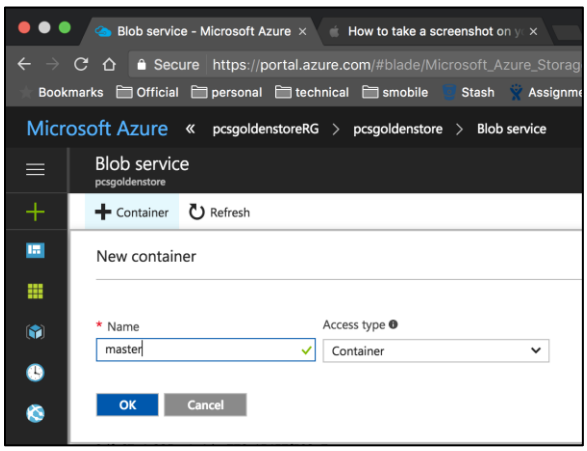
4. Click **New** and create a storage account named 'pcsgoldenstore' under the resource group named 'pcsgoldenstoreRG'.

Figure 12: Storage Account - pcsgoldenstore



5. Inside the pcsgoldenstore storage account, click on **Blobs** and create a container with access type as 'Container' named 'master'.

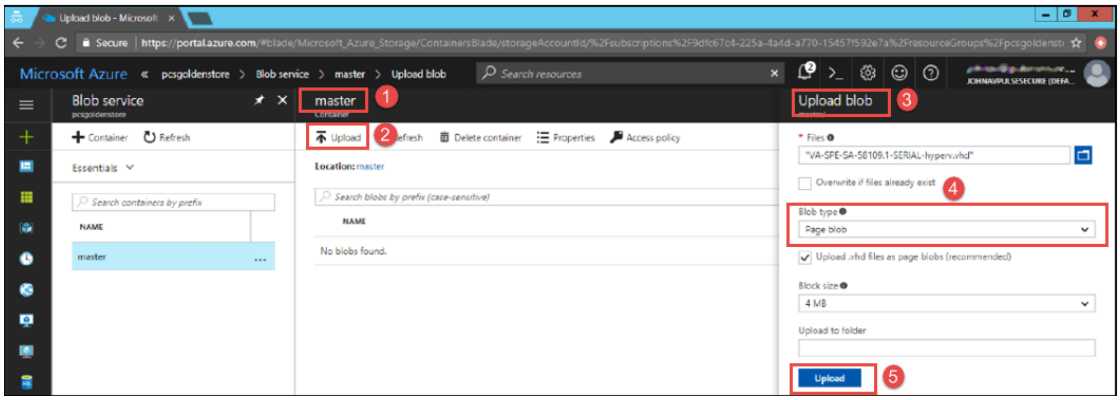
Figure 13: Container master



© 2019 Pulse Secure, LLC. All rights reserved

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

Figure 14: Upload Pulse Connect Secure Virtual Appliance Image



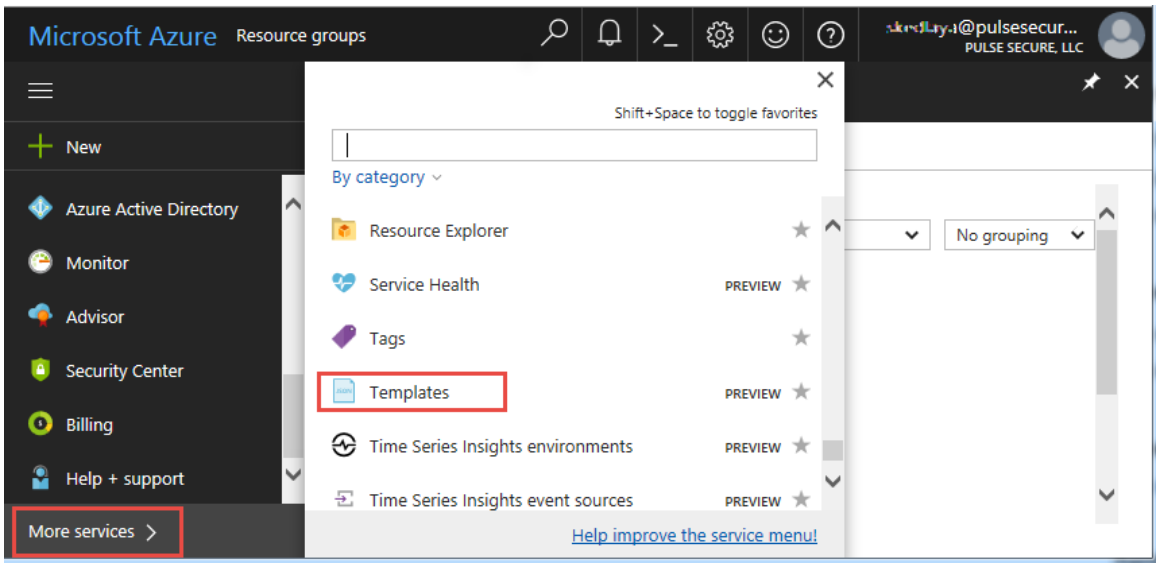
Upload Azure Resource Manager Template to Azure Account

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 two sample Azure template files each for three NIC cards and two NIC cards, namely "pulsesecure-pcs-3-nics.zip" and "pulsesecure-pcs-3-nics-existing-vnet.zip", and "pulsesecure-pcs-2-nics.zip" and "pulsesecure-pcs-2-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-pcs-3-nics.zip** file / **pulsesecure-pcs-2-nics.zip** file to get **azuredeploy.json.**
2. Log in to the Azure portal.
3. Click on **More services** and select **Templates**.

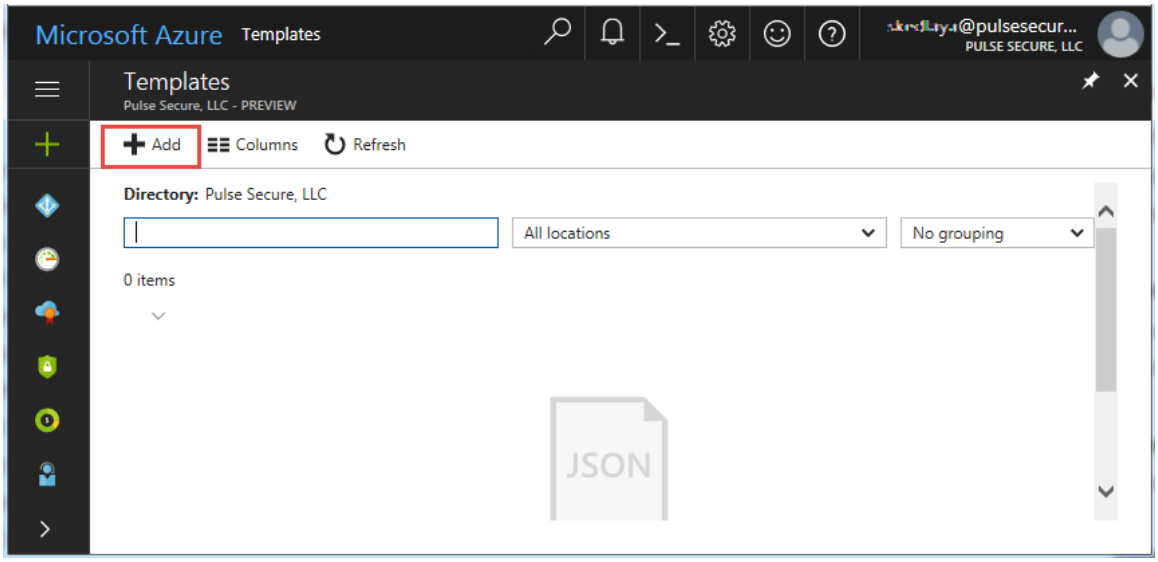
Figure 15: Templates



© 2019 Pulse Secure, LLC. All rights reserved

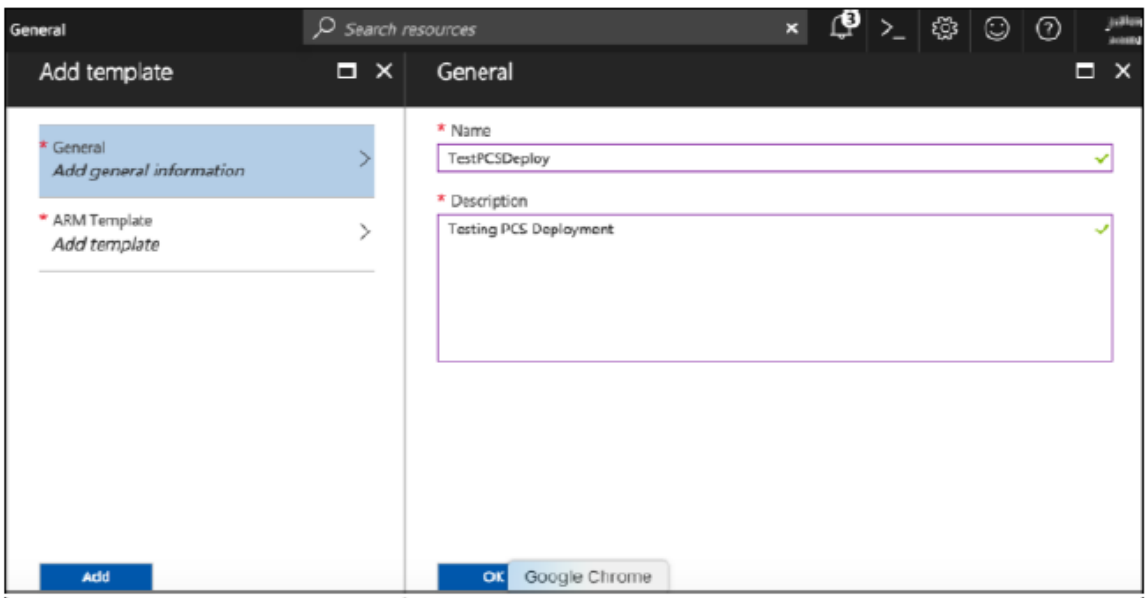
4. In the Templates page, click **Add** to add template.

Figure 16: Add Template



5. Provide a suitable name and description for the template.

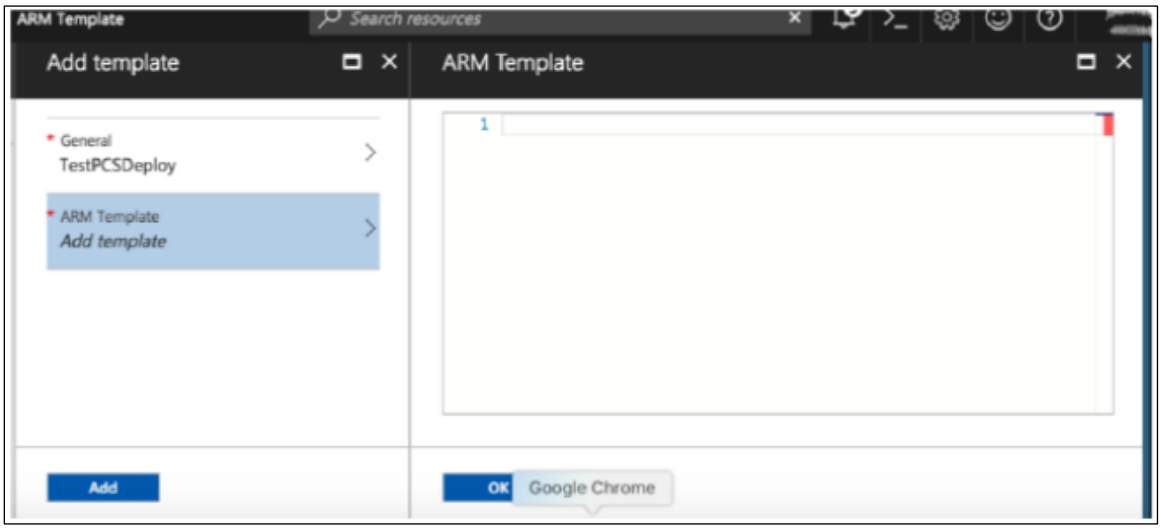
Figure 17: Template – General Information



© 2019 Pulse Secure, LLC. All rights reserved

6. Copy the contents of **azuredeploy.json** and paste it in the template section.

Figure 18: Add ARM Template



Deploying Pulse Connect Secure on Azure using Azure Portal

Before proceeding with the deployment, refer the following sections:

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

Pulse Connect Secure can be deployed on:

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

Deploying PCS on New Virtual Network

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

Deployment on VM with Three NIC Cards

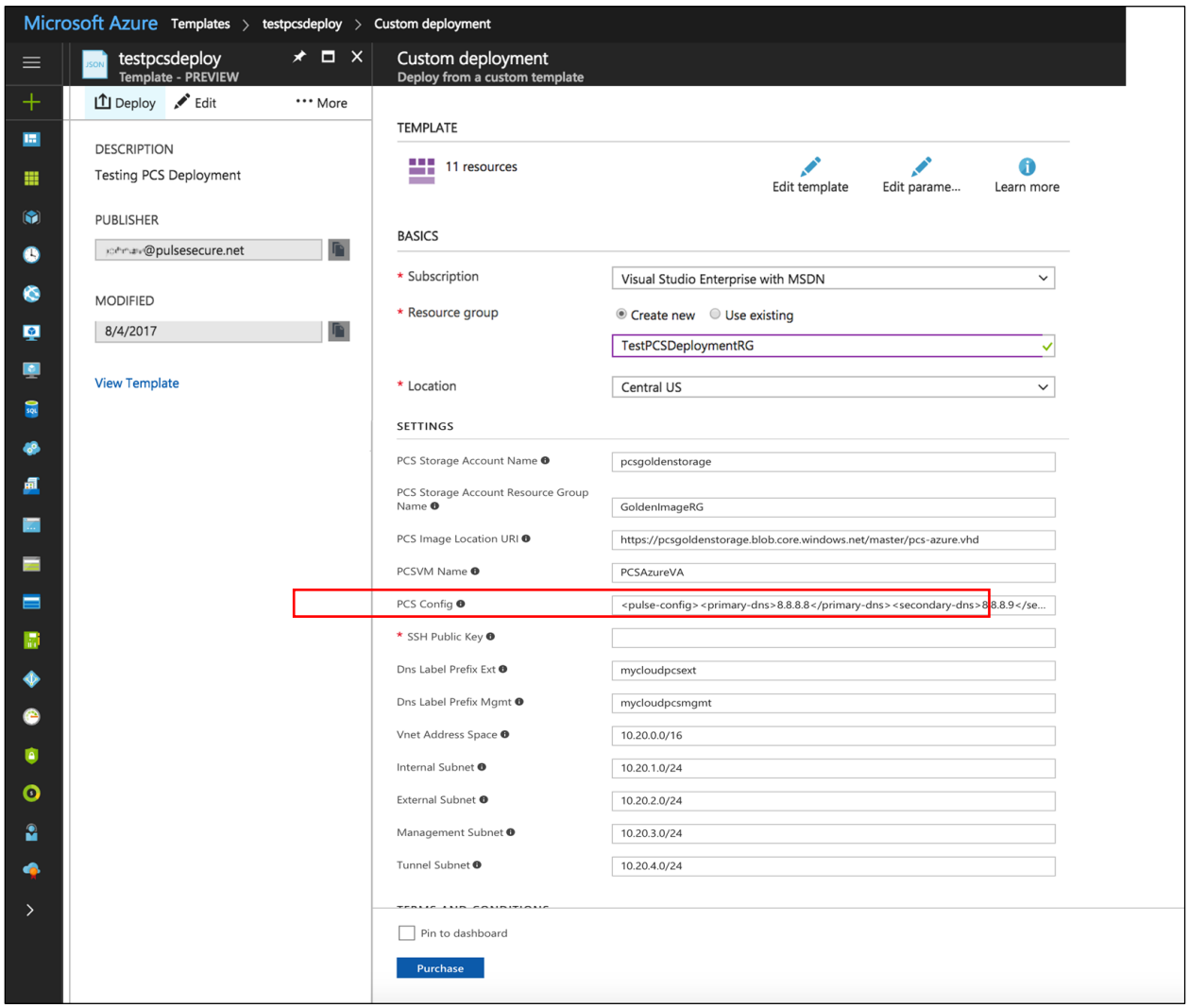
To deploy PCS on Azure using the Azure portal, do the following:

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**".

© 2019 Pulse Secure, LLC. All rights reserved

Figure 19: 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 Connect Secure needs to be deployed
   - **Location:** Region where resource group needs to be created
   - **PCS Storage Account Name:** Storage account name where the Pulse Connect Secure Virtual Appliance image is available
   - **PCS Storage Account Resource Group:** Resource group of where the Pulse Connect Secure Virtual Appliance image is copied
   - **PCS Image Location URI:** URI to Pulse Connect Secure Virtual Appliance Image
   - **PCSVM Name:** Name of the Pulse Connect Secure Virtual instance
   - **PCS Config:** Provisioning parameters in an XML format. Refer the section 'Pulse Connect Secure Provisioning Parameters'
   - **SSH Public Key:** This key is used to access PCS 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:

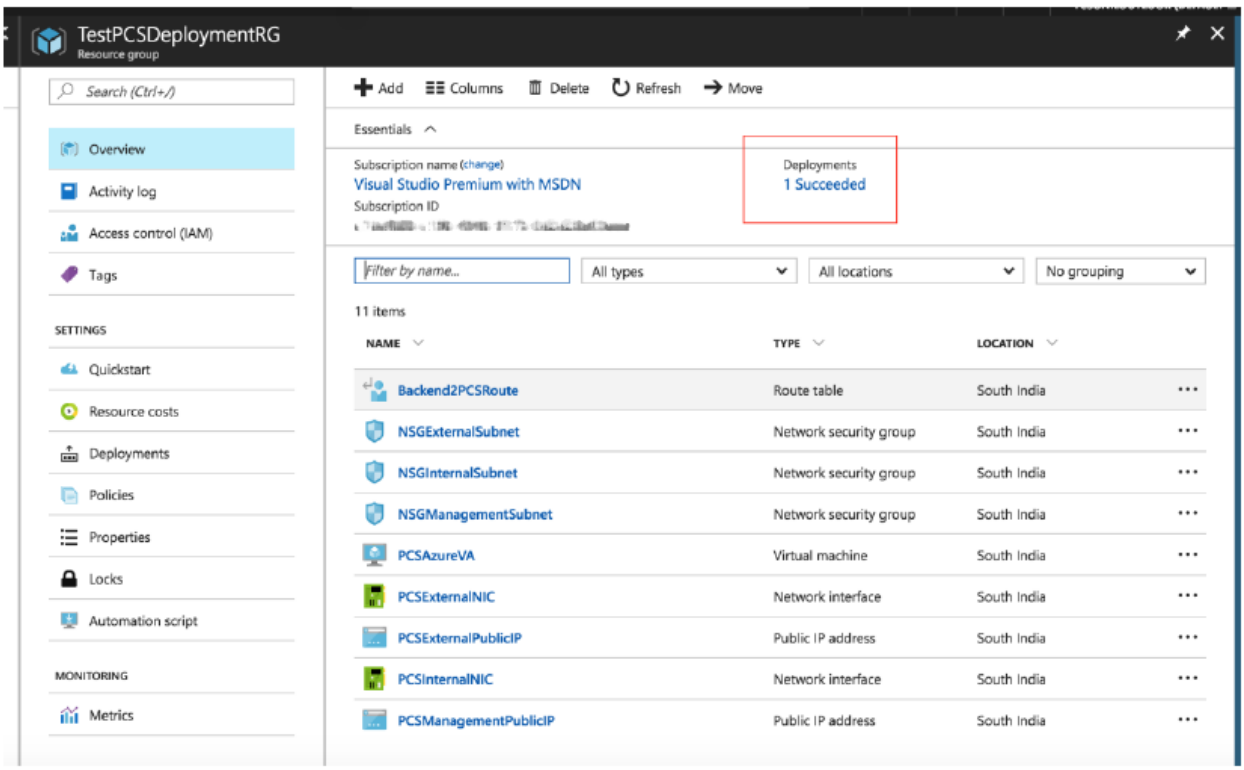
   For Windows: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows
   For MacOS and Linux: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/mac-create-ssh-keys

© 2019 Pulse Secure, LLC. All rights reserved

- **DNS Label Prefix Ext:** Prefix for the external interface DNS label
- **DNS Label Prefix Mgmt:** Prefix for the management interface DNS label
- **Vnet Address Space:** Virtual network address space
- **Internal Subnet:** Subnet from which Pulse Connect Secure internal interface needs to lease IP
- **External Subnet:** Subnet from which Pulse Connect Secure external interface needs to lease IP
- **Management Subnet:** Subnet from which Pulse Connect Secure management interface needs to lease IP
- **Tunnel Subnet:** Subnet which will be configured as tunnel IP pool in the Pulse Connect Secure VPN Profile

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

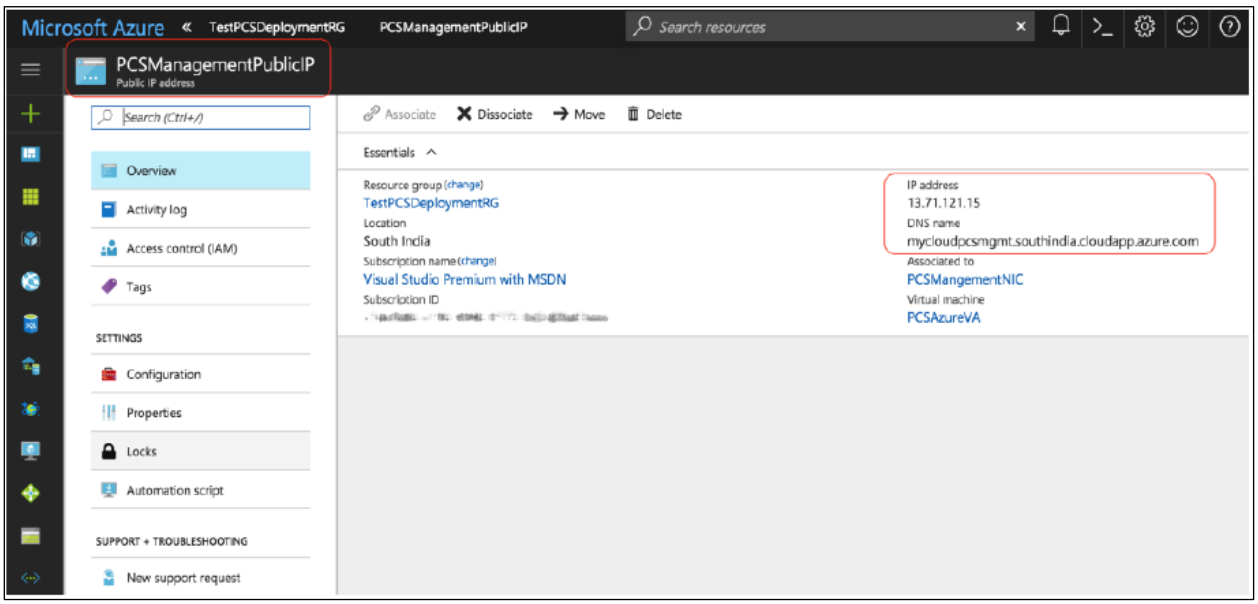
Figure 20: Deployment Succeeded



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

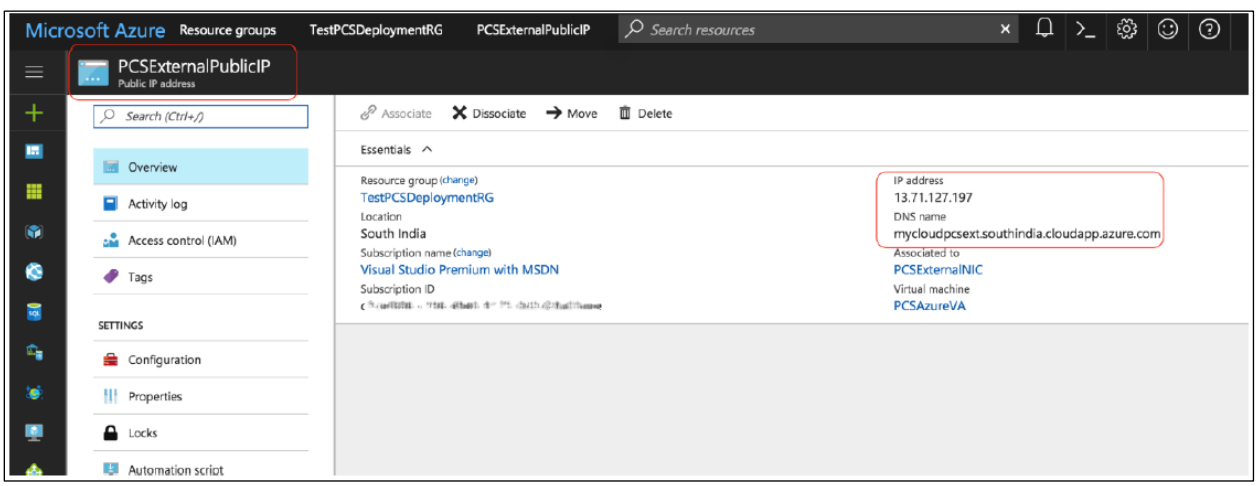
© 2019 Pulse Secure, LLC. All rights reserved

Figure 21: PCS Management Public IP



7. Click **PCS External Public IP** and note down the **PCS External Public IP and DNS name (FQDN)** to access PCS for end user page.

Figure 22: PCS External 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.

© 2019 Pulse Secure, LLC. All rights reserved

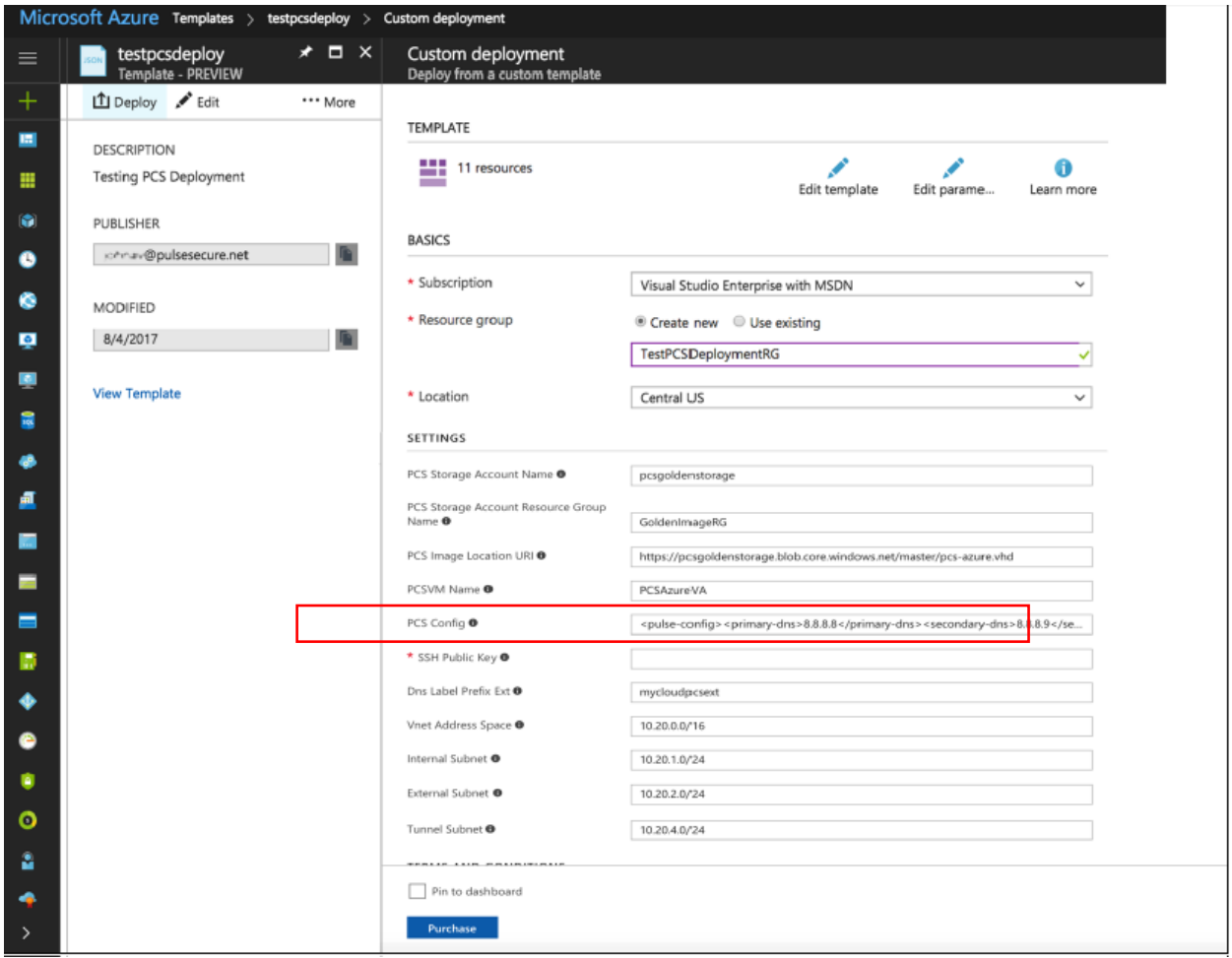
Deployment on VM with Two NIC Cards

To deploy Pulse Connect 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 23: 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 Connect Secure needs to be deployed
   - **Location:** Region where resource group needs to be created
   - **PCS Storage Account Name:** Storage account name where the Pulse Connect Secure Virtual Appliance image is available
   - **PCS Storage Account Resource Group:** Resource group of where the Pulse Connect Secure Virtual Appliance image is copied
   - **PCS Image Location URI:** URI to Pulse Connect Secure Virtual Appliance Image
   - **PCSVM Name:** Name of the Pulse Connect Secure Virtual instance

© 2019 Pulse Secure, LLC. All rights reserved

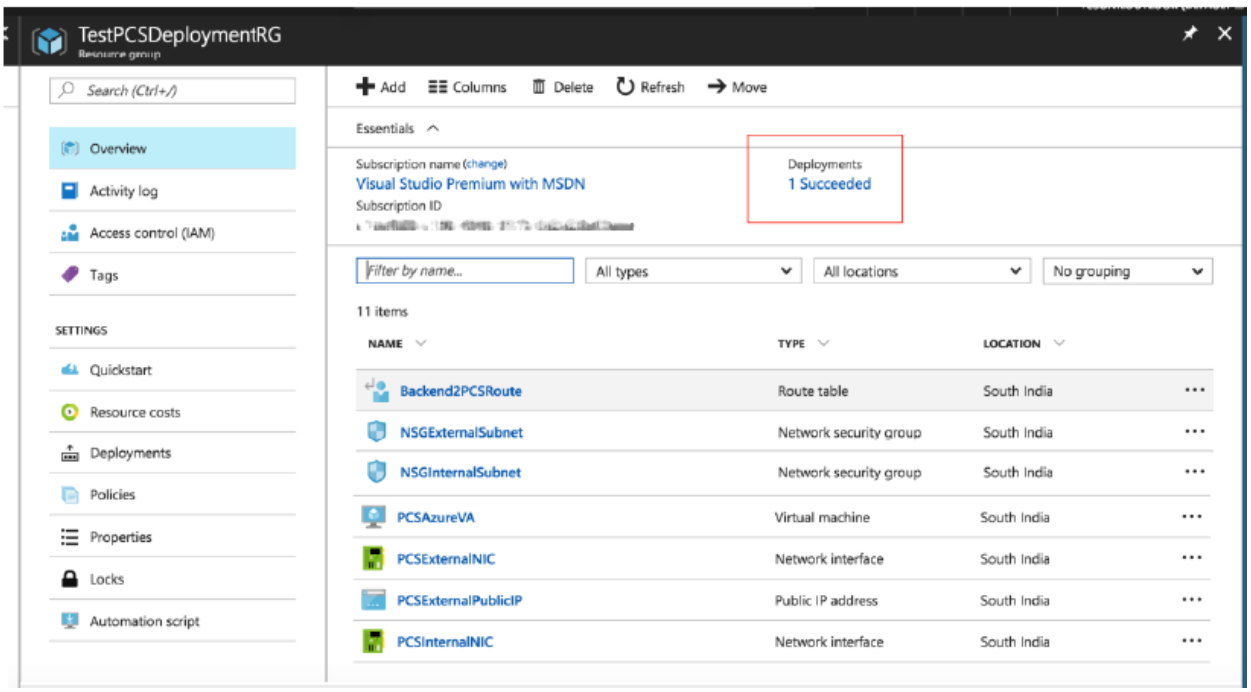
- **PCS Config:** Provisioning parameters in an XML format. Refer the section '[Pulse Connect Secure Provisioning Parameters]'
- **SSH Public Key:** This key is used to access PCS via SSH. The SSH keys are generated using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. For details about generating 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 Ext:** Prefix for the external interface DNS label
- **Vnet Address Space:** Virtual network address space
- **Internal Subnet:** Subnet from which Pulse Connect Secure internal interface needs to lease IP
- **External Subnet:** Subnet from which Pulse Connect Secure external interface needs to lease IP
- **Tunnel Subnet:** Subnet which will be configured as tunnel IP pool in the Pulse Connect Secure VPN Profile

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

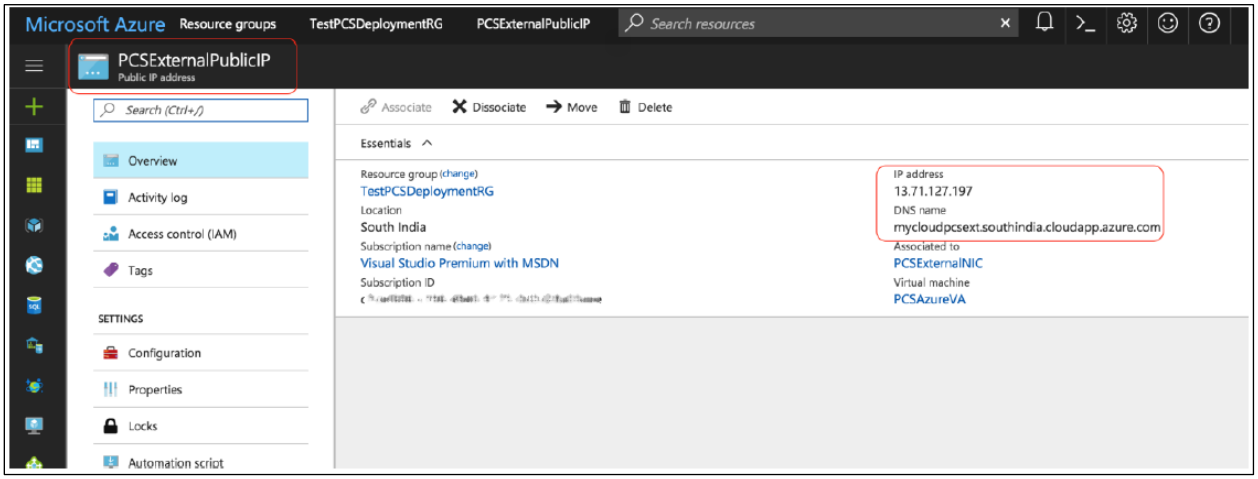


5. Go to the resource group in which the Pulse Connect Secure Virtual Appliance was deployed to see the resources created.

© 2019 Pulse Secure, LLC. All rights reserved

6. Click **PCS External Public IP** and note down the **PCS External Public IP and DNS name (FQDN)** to access PCS for end user page.

Figure 25: PCS External 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.

© 2019 Pulse Secure, LLC. All rights reserved

Deploying PCS on an Existing Virtual Network

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

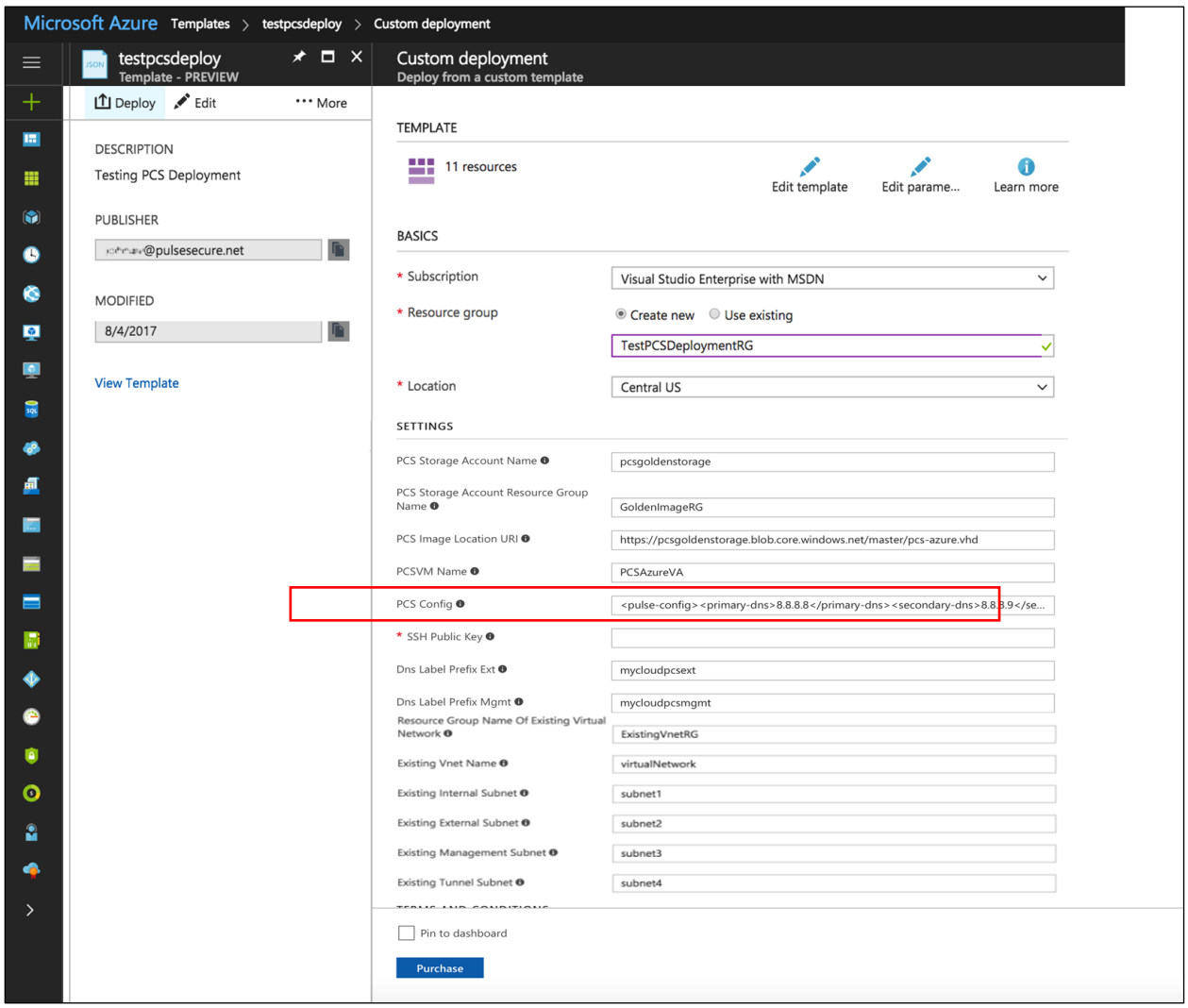
Deployment on VM with Three NIC Cards

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

1. Select the template file "pulsesecure-pcs-3-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 Three NIC Cards – Existing Virtual Network



© 2019 Pulse Secure, LLC. All rights reserved

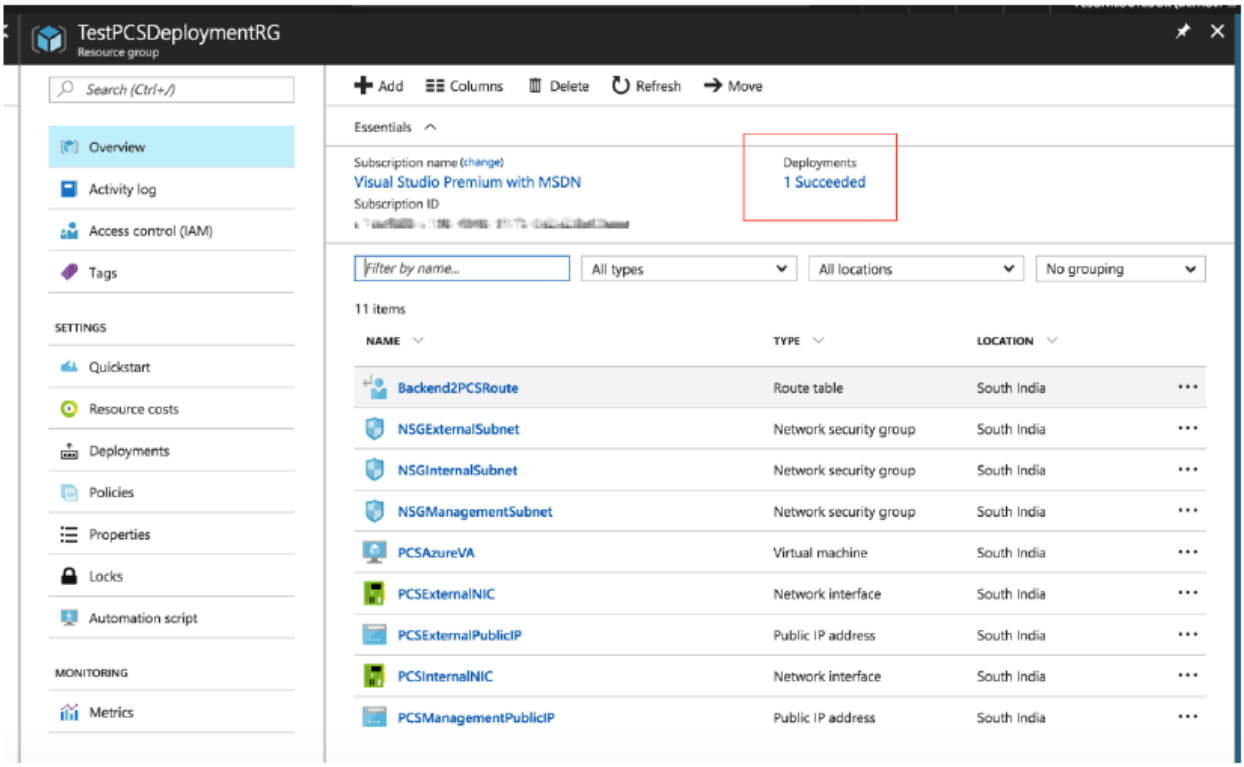
2. Fill or modify the following parameters:
   - **Resource group:** Specify the resource group name in which Pulse Connect Secure needs to be deployed
   - **Location:** Region where resource group needs to be created
   - **PCS Storage Account Name:** Storage account name where the Pulse Connect Secure Virtual Appliance image is available
   - **PCS Storage Account Resource Group:** Resource group of where the Pulse Connect Secure Virtual Appliance image is copied
   - **PCS Image Location URI:** URI to Pulse Connect Secure Virtual Appliance Image
   - **PCS VM Name:** Name of the Pulse Connect Secure Virtual instance
   - **PCS Config:** Provisioning parameters in XML format. Refer 'Pulse Connect Secure Provisioning Parameters'
   - **SSH Public Key:** This key is used to access PCS 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 Ext:** Prefix for the external interface DNS label
   - **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 Connect Secure internal interface needs to lease IP
   - **Existing External Subnet:** Subnet from which the Pulse Connect Secure external interface needs to lease IP
   - **Existing Management Subnet:** Subnet from which the Pulse Connect Secure management interface needs to lease IP
   - **Existing Tunnel Subnet:** Subnet which will be configured as the tunnel IP pool in the Pulse Connect Secure VPN Profile
3. Agree to the Azure licensing terms and click **Purchase**.

© 2019 Pulse Secure, LLC. All rights reserved

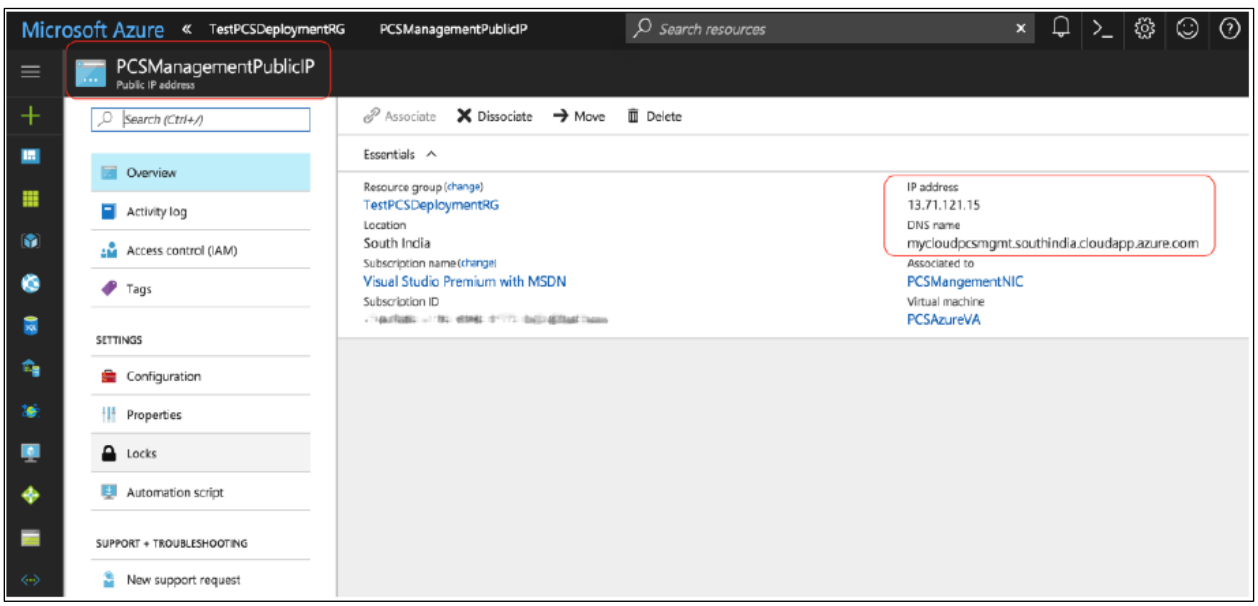
4. Watch for the deployment succeeded message after 3 to 5 minutes.

Figure 27: Deployment Succeeded



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

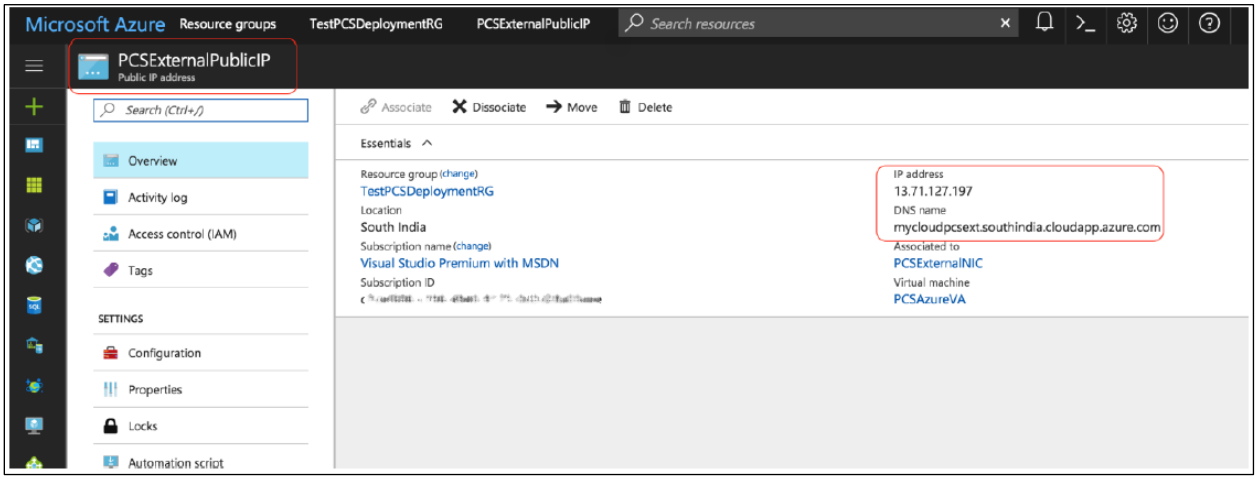
Figure 28: PCS Management Public IP



© 2019 Pulse Secure, LLC. All rights reserved

7. Click **PCS External Public IP** and note down the **PCS External Public IP and DNS name (FQDN)** to access PCS for end user page.

Figure 29: PCS External 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.

© 2019 Pulse Secure, LLC. All rights reserved

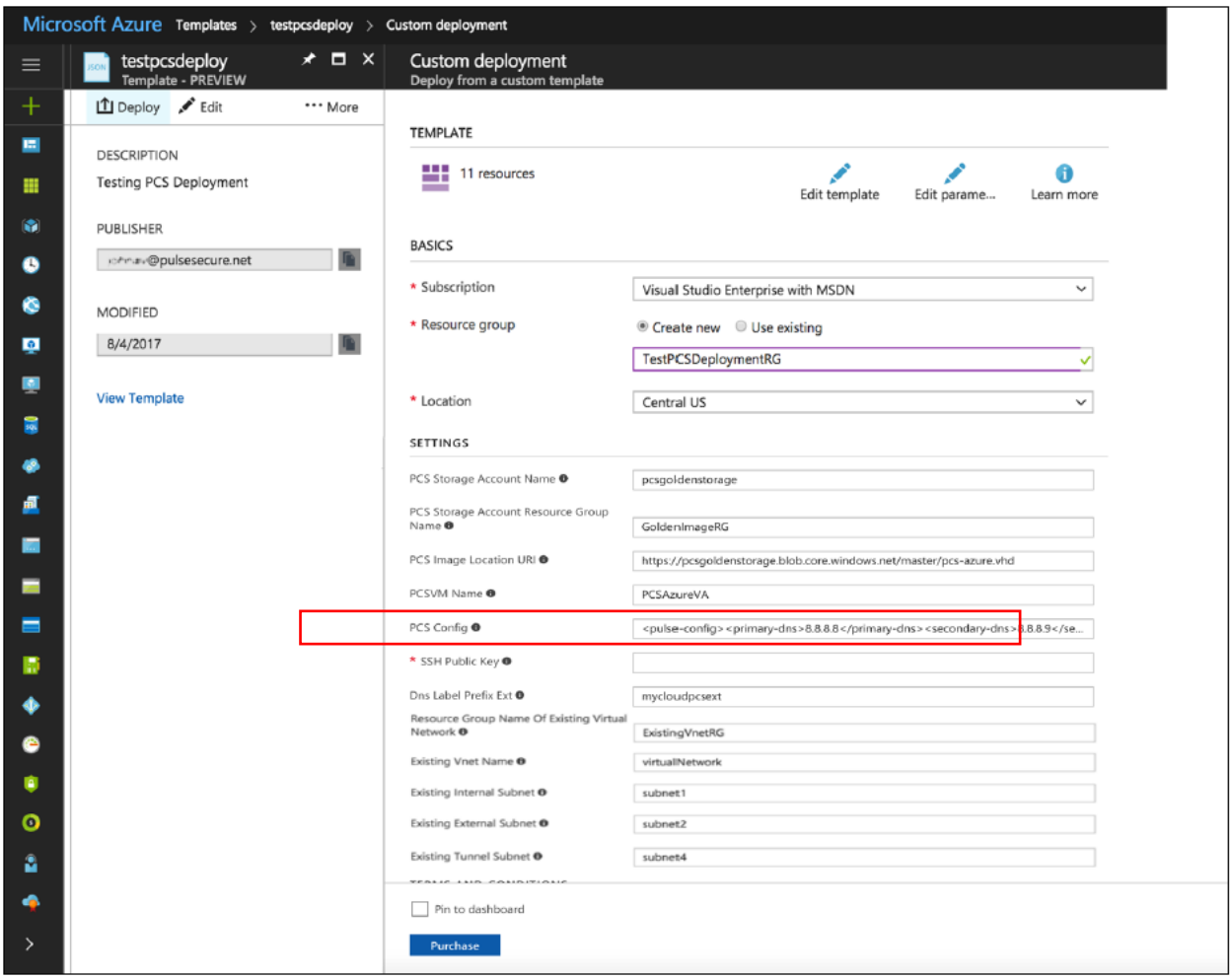
Deployment on VM with Two NIC Cards

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

1. Select the template file "pulsesecure-pcs-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 30: Custom Deployment on VM with Two NIC Cards – Existing Virtual Network



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

© 2019 Pulse Secure, LLC. All rights reserved

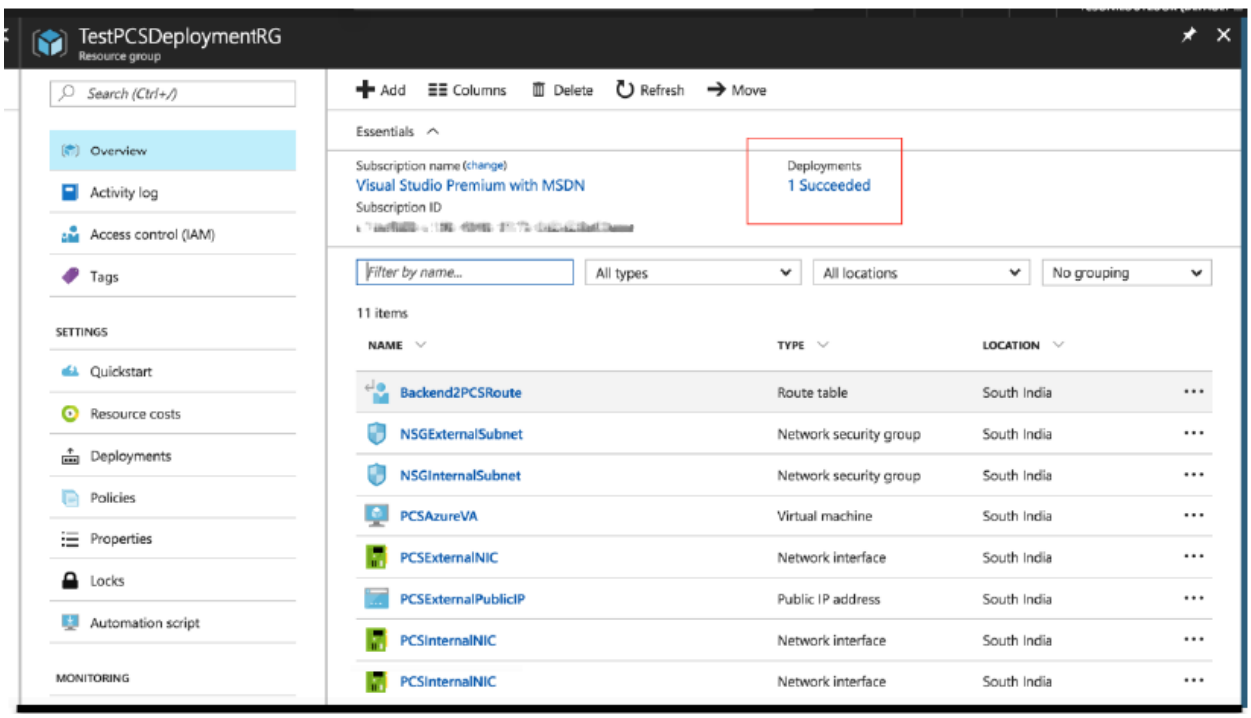
- **PCS Config:** Provisioning parameters in XML format. Refer 'Pulse Connect Secure Provisioning Parameters'
- **SSH Public Key:** This key is used to access PCS via SSH. The SSH keys are generated using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, 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 Ext:** Prefix for the external 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 Connect Secure internal interface needs to lease IP
- **Existing External Subnet:** Subnet from which the Pulse Connect Secure external interface needs to lease IP
- **Existing Tunnel Subnet:** Subnet which will be configured as the tunnel IP pool in the Pulse Connect Secure VPN Profile

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

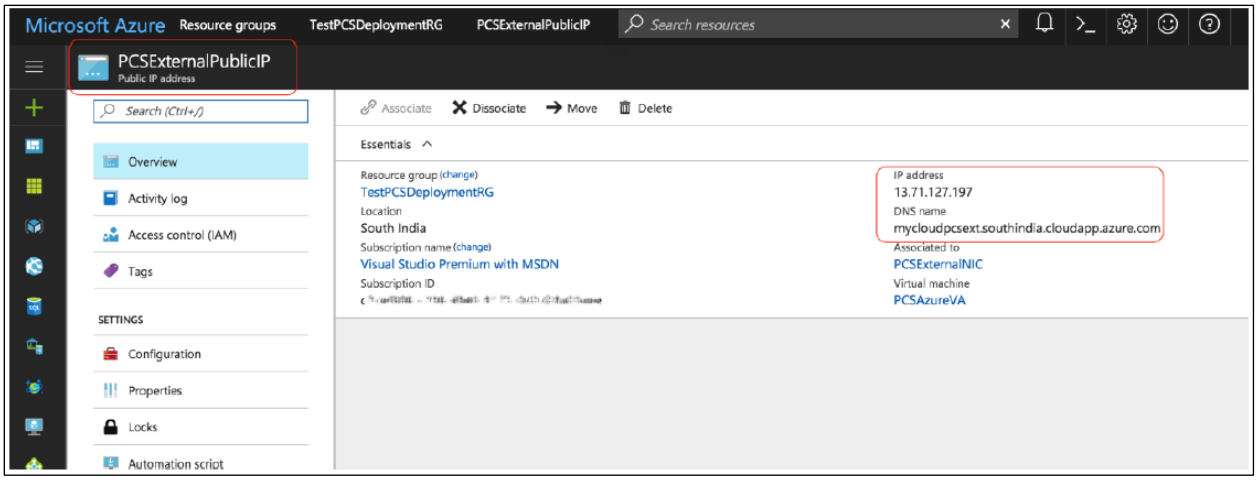
Figure 31: Deployment Succeeded



© 2019 Pulse Secure, LLC. All rights reserved

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

Figure 32: PCS External 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.

© 2019 Pulse Secure, LLC. All rights reserved

Deploying Pulse Connect Secure on Azure using Azure CLI

Before proceeding with the deployment, refer Upload Pulse Connect 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-pcs-azure-psa-v-<releaseno>-<buildno>-package.zip** file.
3. Unzip the file and look for the **pulsesecure-pcs-3-nics.zip** file. Unzip the file to get **azuredeploy.json**
4. Ensure that parameters section has correct default values:
   - **PCS Storage Account Name:** Storage account name where the Pulse Connect Secure Virtual Appliance image is available
   - **PCS Storage Account Resource Group:** Resource group where the Pulse Connect Secure Virtual Appliance image is copied
   - **PCS Image Location URI:** URI to the Pulse Connect Secure Virtual Appliance Image
   - **PCS VM Name:** Name of the Pulse Connect Secure Virtual instance
   - **PCS Config:** Provisioning parameters in an XML format. Refer "Pulse Connect Secure Provisioning Parameters"
   - **DNS Label Prefix Ext:** Prefix for the external interface DNS label
   - **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 Connect Secure internal interface needs to lease IP
5. To deploy Pulse Connect 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:

```
$ az login
```

```
: Azure.:Desktop Azure az login
To sign in, use a web browser to open the page https://aka.ms/devicelogin and enter the code GDGZ7EE9Z to authenticate.
[
  {
    "cloudName": "AzureCloud",
    "id": "cloud586-c196-c196-c196-0d2d28a03eee",
    "isDefault": true,
    "name": "Visual Studio Premium with MSDN",
    "state": "Enabled",
    "tenantId": "cloudb2c-a0fd-a0fd-a0fd-72c2932cd84d",
    "user": {
      "name": "Azure@outlook.com",
      "type": "user"
    }
  }
]
: Azure.:Desktop : Azure$
```

```
$ az group create -l southindia -n TestPCSDeploymentRG
```

© 2019 Pulse Secure, LLC. All rights reserved

```
TestPCS.:Desktop TestPCS$ az group create -l southindia -n TestPCSDeploymentRG
{
  "id": "/subscriptions/TestPCS6-9175-9175-9175-southindiaee/resourceGroups/TestPCSDeploymentRG",
  "location": "southindia",
  "managedBy": null,
  "name": "TestPCSDeploymentRG",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
TestPCS.:Desktop TestPCS$
```

$ az group deployment create -g TestPCSDeploymentRG --template-file azuredeploy.json

```
      {
        "id": null,
        "namespace": "Microsoft.Compute",
        "registrationState": null,
        "resourceTypes": [
          {
            "aliases": null,
            "apiVersions": null,
            "locations": [
              "southindia"
            ],
            "properties": null,
            "resourceType": "virtualMachines"
          }
        ]
      }
    ],
    "provisioningState": "Succeeded",
    "template": null,
    "templateLink": null,
    "timestamp": "2017-08-06T17:19:20.227838+00:00"
  },
  "resourceGroup": "TestPCSDeploymentRG"
}
```

© 2019 Pulse Secure, LLC. All rights reserved

Pulse Connect Secure Provisioning Parameters

Provisioning parameters are those parameters which are required during the deployment of a virtual appliance. Pulse Connect 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 Connect Secure
2	secondary-dns	IP address	Secondary DNS for Pulse Connect Secure
3	wins-server	IP address	Wins server for Pulse Connect Secure
4	dns-domain	string	DNS domain of Pulse Connect 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 Connect 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 Connect Secure configuration can be found. During provisioning, Pulse Connect 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 Connect Secure through site to site VPN between Azure and corporate data center
11	config-data	string	base64 encoded XML based Pulse Connect 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', PCS will be deployed as a License server.

© 2019 Pulse Secure, LLC. All rights reserved

			If set to 'n', PCS 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.

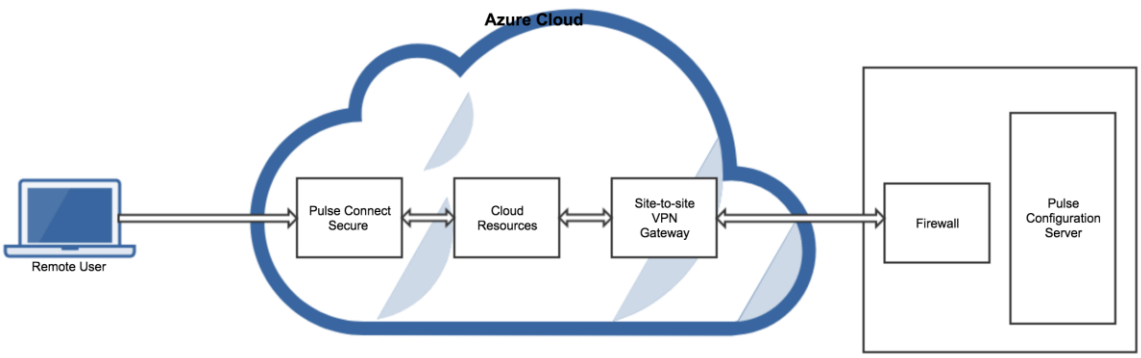
Provisioning Pulse Connect Secure with Predefined Configuration

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

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

  Pulse Connect 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 Connect Secure administrator needs to ensure that site-to-site VPN between Azure to corporate network is already established so that Pulse Connect Secure can access the machine located in the corporate network.
- Pulse Connect Secure administrator provides the configuration data encoded in the base64 encoded xml in the ARM template.

Figure 33: Pulse Configuration Server in Corporate Network



© 2019 Pulse Secure, LLC. All rights reserved

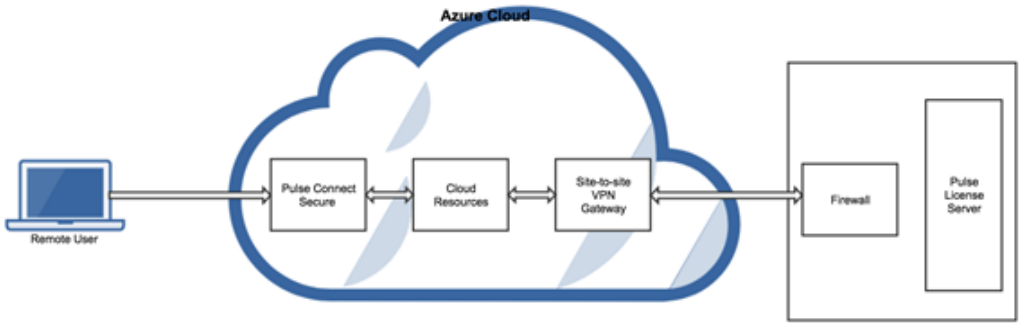
Configuring Licenses on the Pulse Connect Secure Appliance

By default, two-user licenses are provided. To add more licenses, the Pulse Connect 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 34: Pulse License Server in a Corporate Network

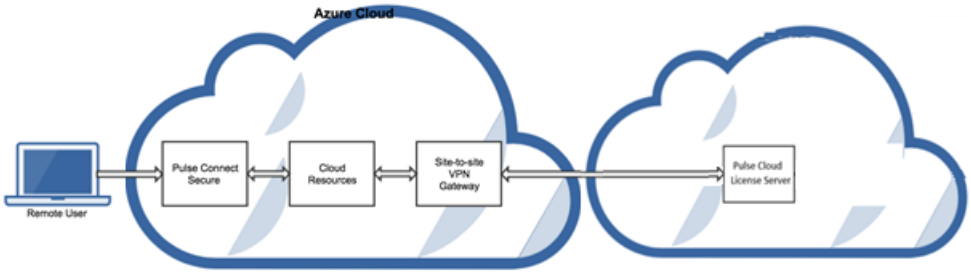


Pulse License Server in Cloud Network

In Pulse Connect Secure 8.3R3, the Pulse Connect 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 PCS admin console. The PCS also periodically sends heartbeat messages to PCLS for auditing purposes.
The Authentication code can also be specified in the ARM template. When PCS comes up, it automatically fetches the Authentication code.

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

Figure 35: Pulse License Server in Cloud Network



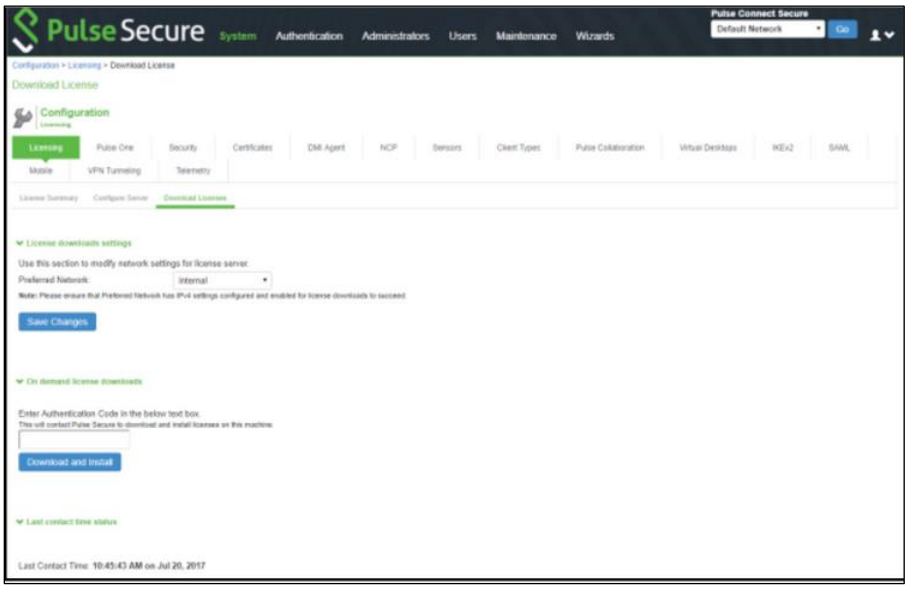
© 2019 Pulse Secure, LLC. All rights reserved

Adding Authentication Code in PCS 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 36: 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 PCSConfig 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>",
```

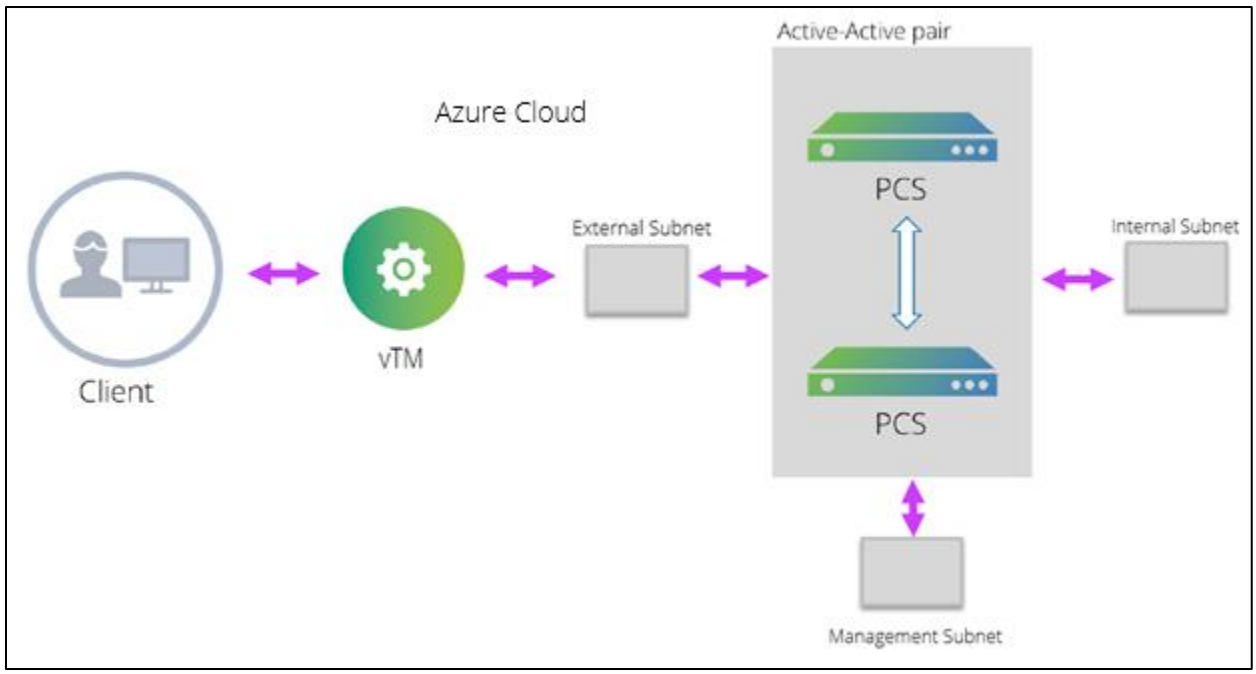
For details about the license configuration, refer to License Configuration Guide.

© 2019 Pulse Secure, LLC. All rights reserved

Deploying PCS Active-Active Cluster using Virtual Traffic Manager in Microsoft Azure

This section describes deploying PCS A-A cluster with vTM load balancer in Microsoft Azure.

Figure 37: Deploying PCS A-A Cluster Topology Diagram



The deployment process involves the following steps:

- Deploying Two PCS EC2 instances Using ARM Template
- Forming the Active-Active Cluster
- Deploying Virtual Traffic Manager EC2 Instance in the External Subnet of PCS in
- Setting Up and Configuring vTM for External Users

Deploying Two PCS EC2 instances Using ARM Template

PCS can be deployed in Azure using ARM template in a 3-armed model. Based on the need, deploy two PCS instances using the json template from one of the following zip files:

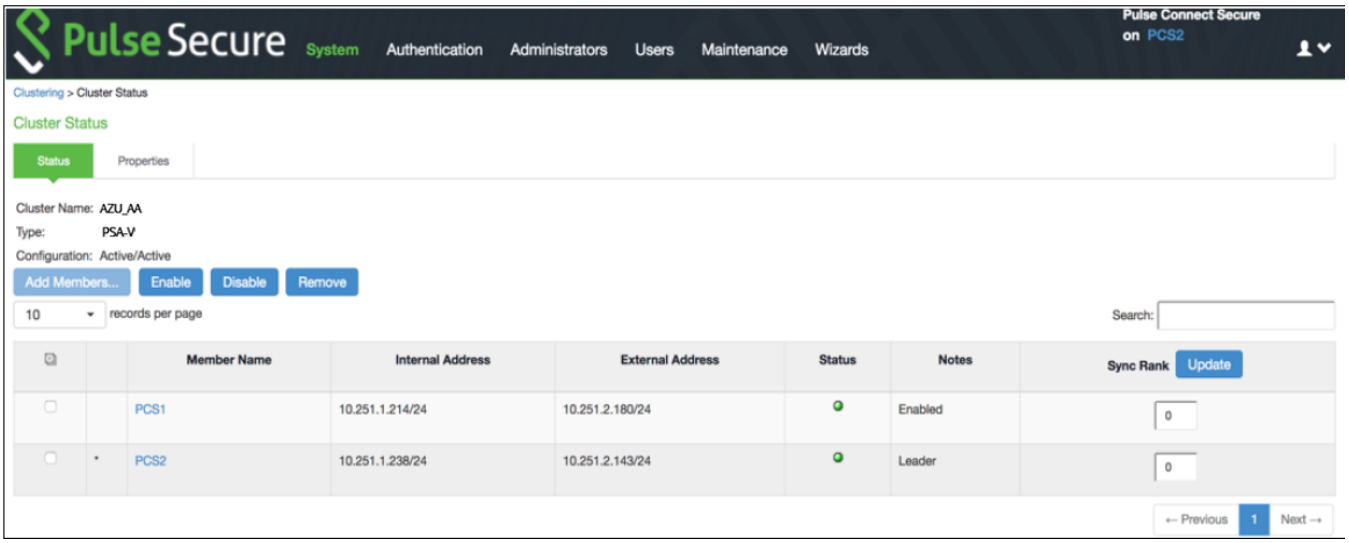
- pulsesecure-pcs-3-nics.zip
- pulsesecure-pcs-3-nics-existing-vnet.zip

Forming the Active-Active Cluster

Once the two PCS instances are initialized, form the Active-Active cluster between them. For details about creating PCS clusters, refer to PCS Administration Guide published in the Pulse Secure Techpubs site.

© 2019 Pulse Secure, LLC. All rights reserved

Figure 38: PCS A-A Cluster Status



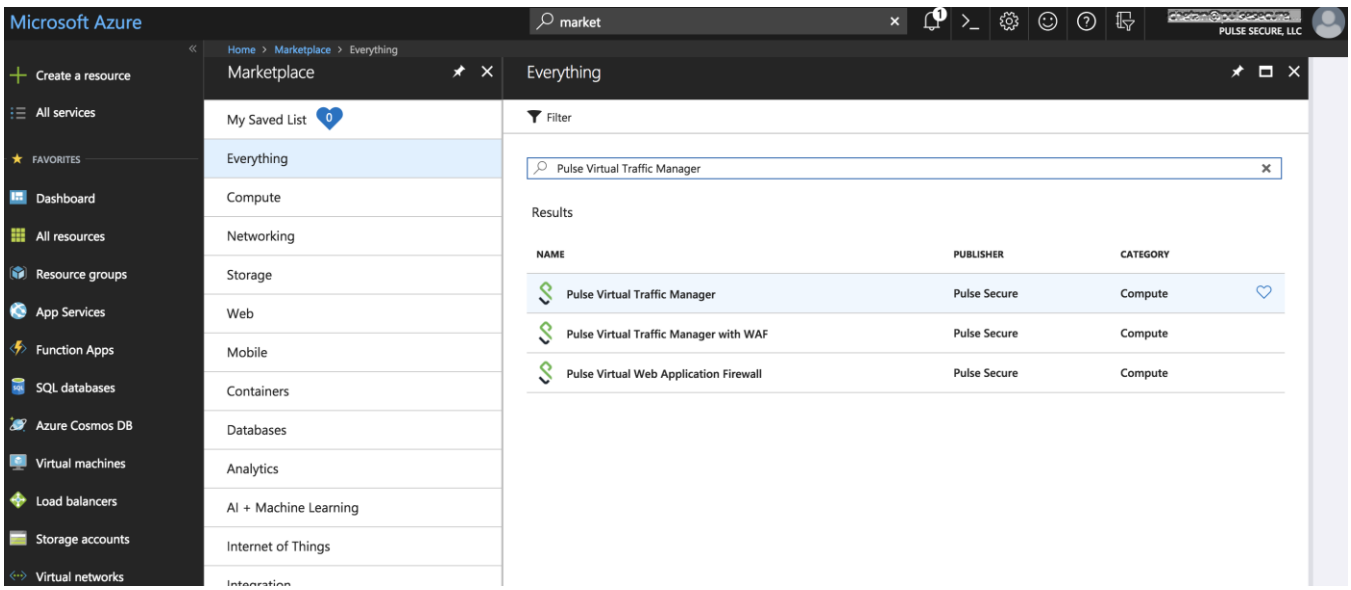
Deploying Virtual Traffic Manager EC2 Instance in the External Subnet of PCS in Microsoft Azure

Virtual Traffic Manager can be deployed through either Azure Marketplace or Azure CLI.
Deploying Virtual Traffic Manager through Marketplace includes the following steps:

To deploy through Marketplace, follow the below steps:

1. Search and select **Pulse Secure vTM** in Azure Marketplace.

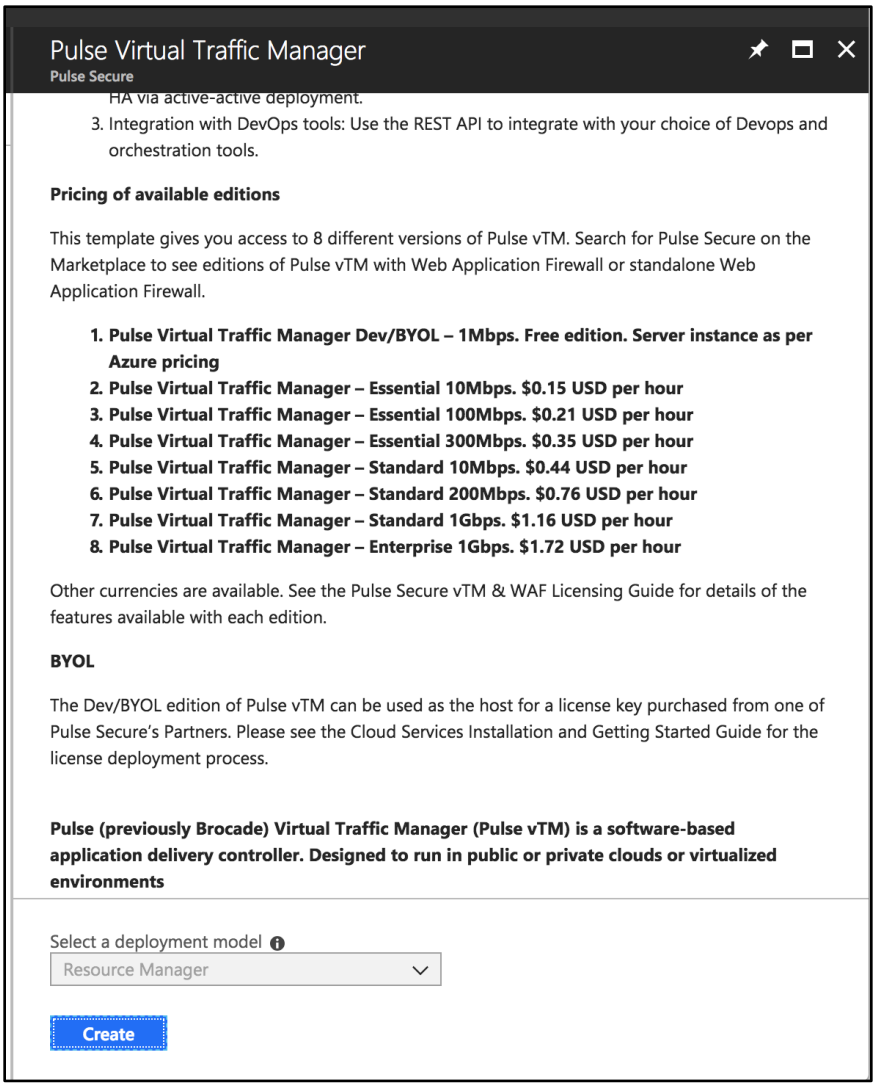
Figure 39: Azure Marketplace > Pulse Secure vTM



© 2019 Pulse Secure, LLC. All rights reserved

2. Select the required deployment model and click **Create**.

Figure 40: vTM Editions Available in Azure Marketplace

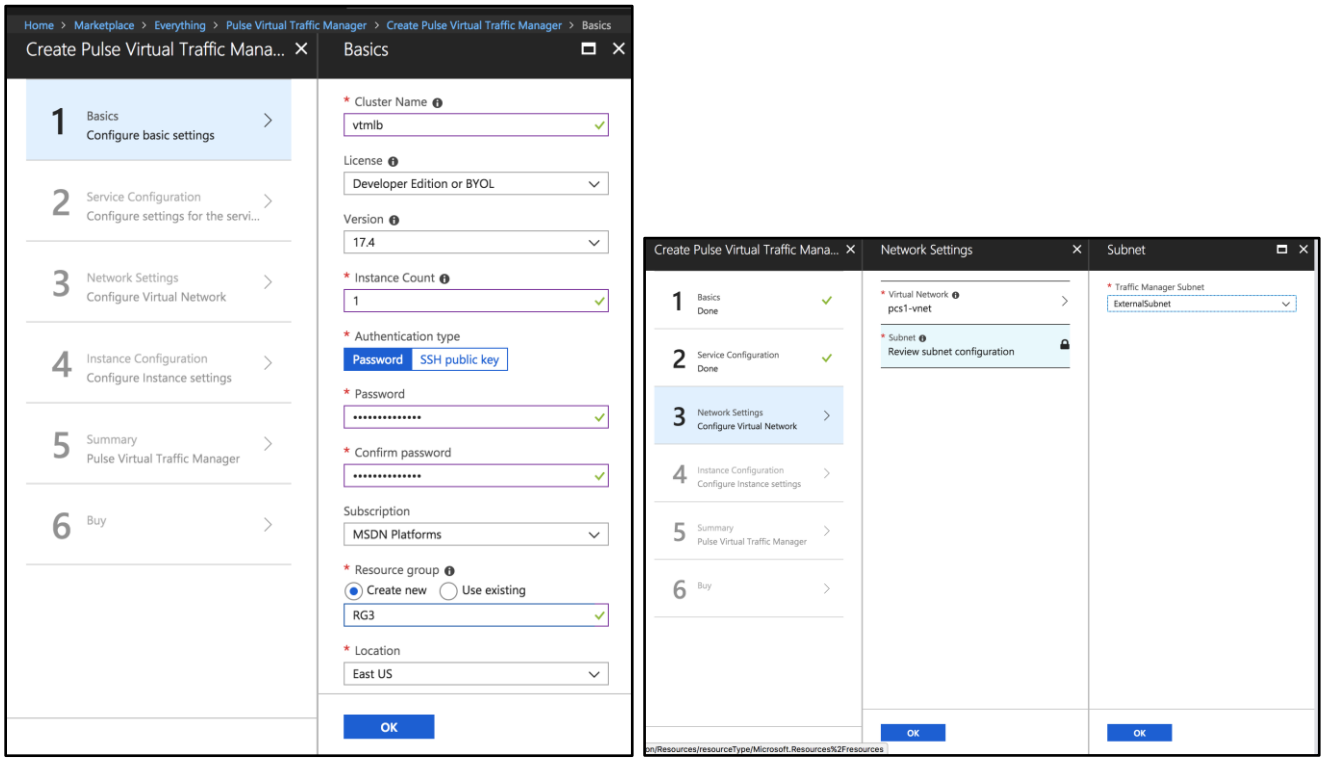


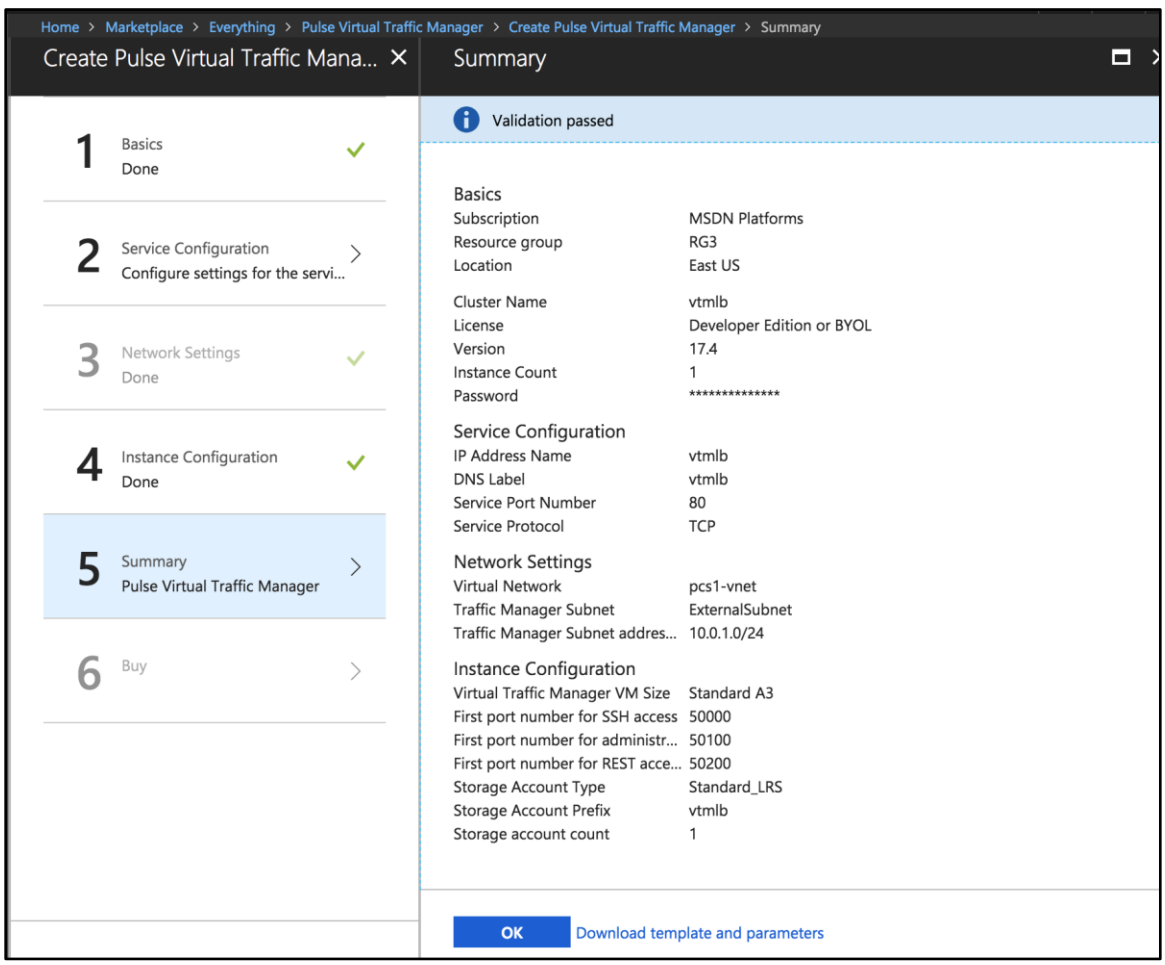
3. In the wizard that follows, provide the required configuration details:
   - Cluster name
   - License type
   - Authentication details
   - Virtual Network and Subnet settings
   - Resource group
   - Location information

   In the **Network Settings** tab, select the Virtual Network and Traffic Manager Subnet matching PCS's Vnet and External Subnet.

© 2019 Pulse Secure, LLC. All rights reserved

Figure 41: Configuration Wizard





© 2019 Pulse Secure, LLC. All rights reserved

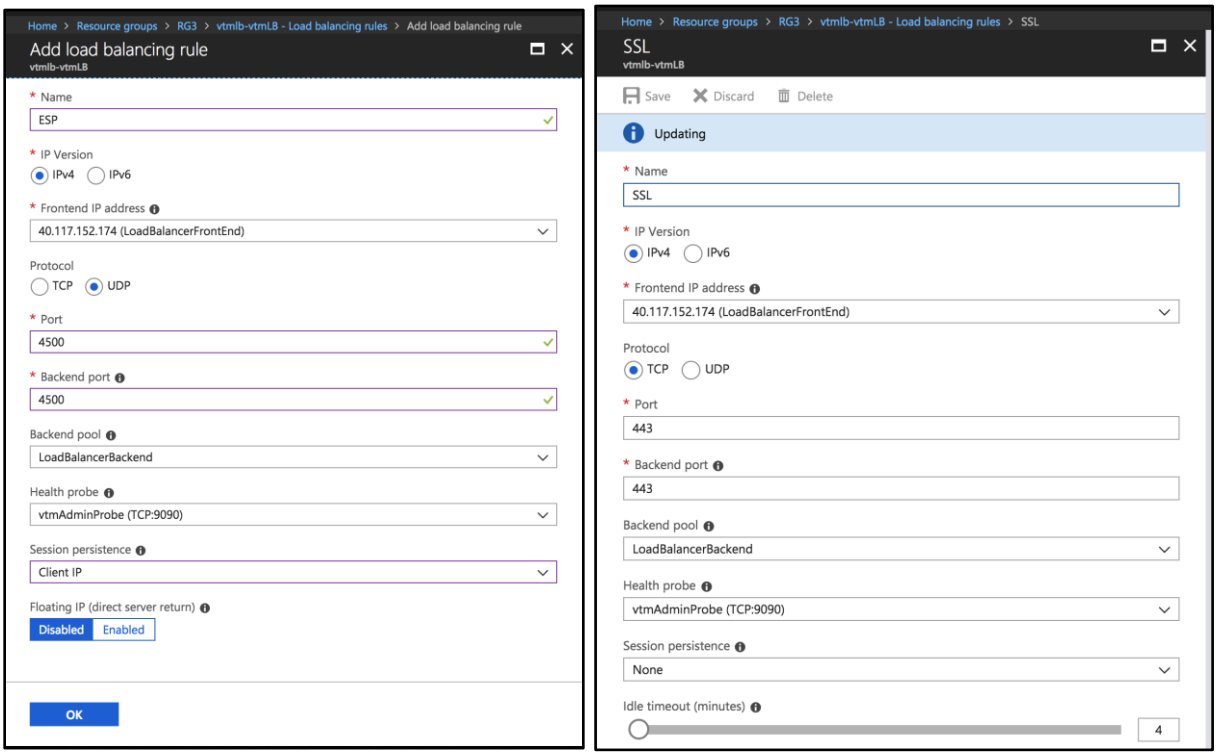
Adding Load Balancing and Inbound Network Security Rules

To manage an additional service in your Traffic Manager cluster, or if the existing service uses multiple ports or protocols, add load balancer and network security rules after creating the cluster.

To add load balancing rule, perform the following steps:

1. Navigate to your resource group.
2. Click the Load Balancer resource name (typically named "<clustername>-vtmLB").
3. From the load balancer settings pane, click **Load balancing rules**.
4. Click **Add**.

Figure 42: Add Load Balancing Rule

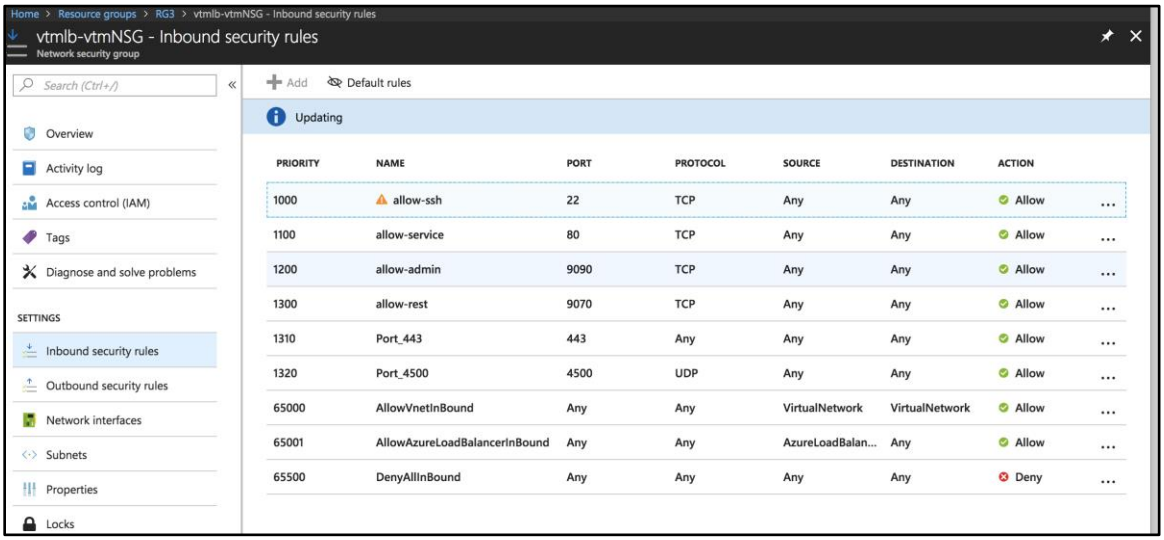


5. Configure the following settings for ESP and SSL traffic modes, and click **OK**:
   - **Name**: Type a descriptive name for this rule.
   - **Protocol**: Select your traffic protocol.
   - **Port**: Enter the port number for your traffic.
   - **Backend Port**: Set to the same value as Port.
   - **Session Persistence**: Select "None".
   - **Idle Timeout (minutes):** Set to a timeout value suitable for your service.
   - **Floating IP (direct server return)**: Select "Disabled".

6. In the resource group, click the name of the Network Security Group resource (typically named "<clustername>-vtmNSG").
7. Click **Inbound Security Rules** and then click **Add**.

© 2019 Pulse Secure, LLC. All rights reserved

Figure 43: Inbound Security Rules



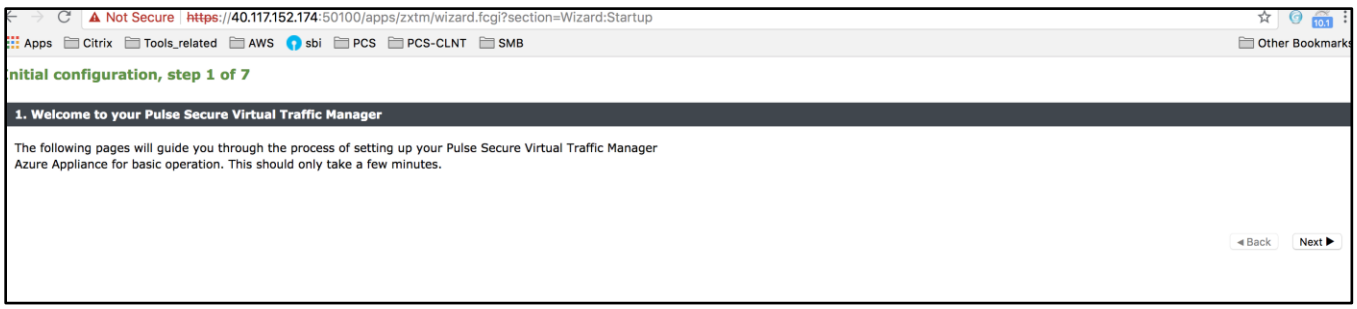
8. Configure the following settings:
   - Name: Type a descriptive name for this rule.
   - Priority: Enter the desired priority number. The higher the priority number, the lower the priority over other rules.
   - Source: Select "Any".
   - Protocol: Select your traffic protocol.
   - Source Port Range: Leave this setting as the default "*".
   - Destination: Select "Any".
   - Destination Port Range: Enter the port number or range for your traffic.
   - Action: Select "Allow".
9. Click OK to save the rule.

Pulse Secure Virtual Traffic Manager Initial Configuration

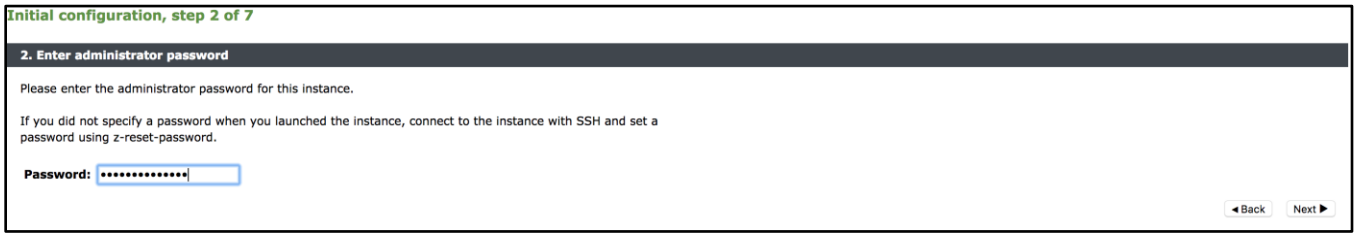
A newly created Virtual Traffic Manager requires some basic information to function normally.
Use the Initial Configuration wizard by entering the URL of the Admin UI into your Web browser. Provide the following details:

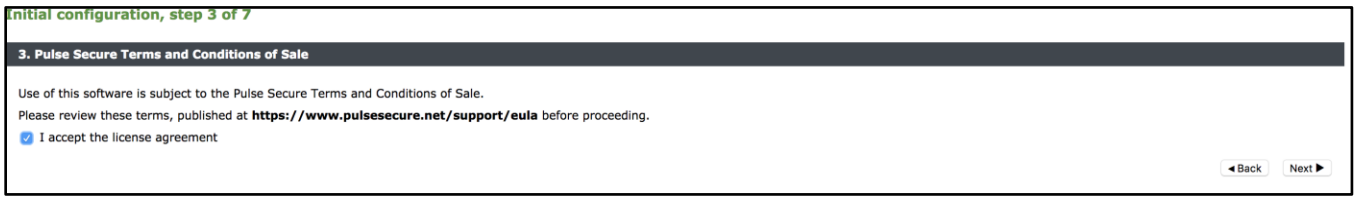
- Administrator password for the instance
- Confirmation to the terms and conditions
- Time zone settings for the appliance
- Login credentials for master admin user to log in to the Administration server and SSH console
- Licensing option

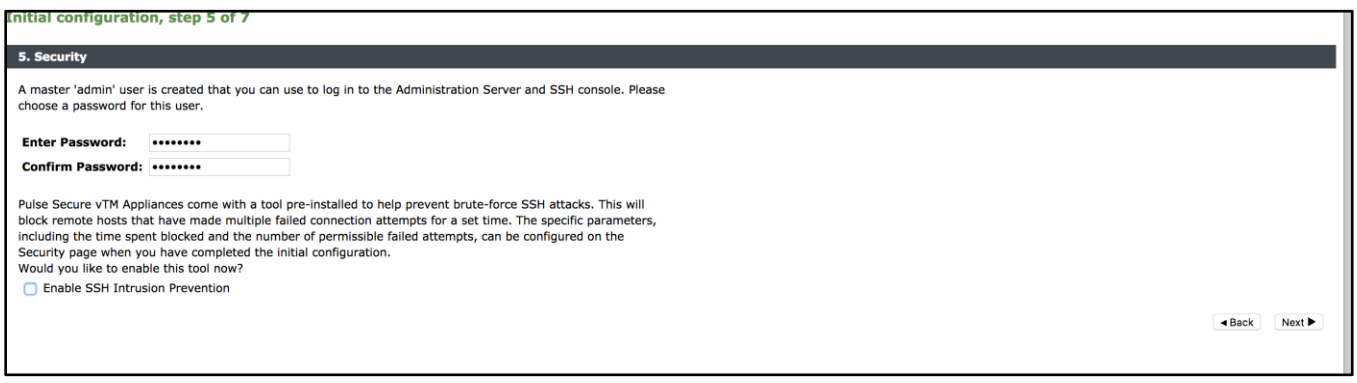
Figure 44: Pulse Secure vTM Initial Configuration Wizard

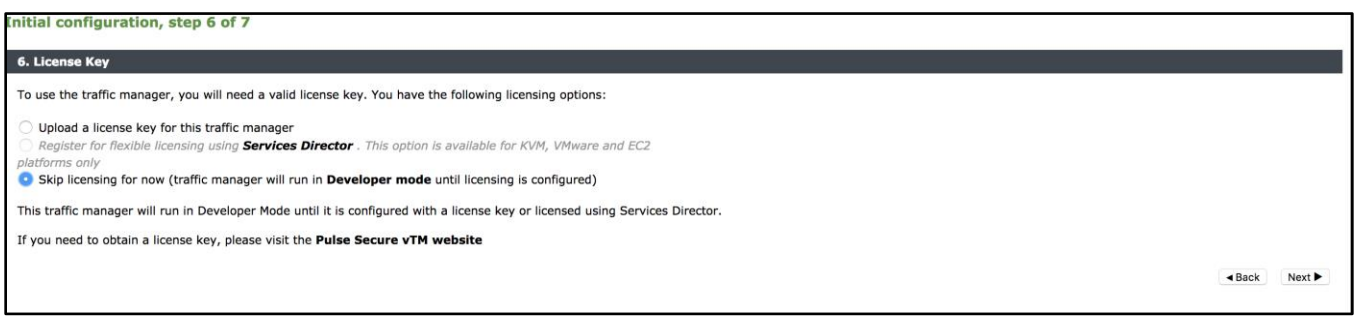


© 2019 Pulse Secure, LLC. All rights reserved









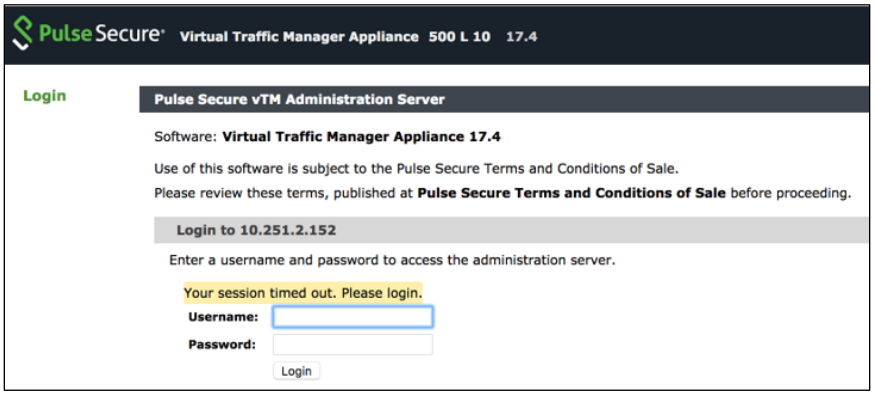
For additional details, and to deploy vTM though Azure CLI, follow the steps in the section “Creating a Traffic Manager Instance on Azure EC2” in Pulse Secure Virtual Traffic Manager: Cloud Services Installation and Getting Started Guide. Make sure that vTM is deployed on the external network of PCS.

© 2019 Pulse Secure, LLC. All rights reserved

Setting Up and Configuring vTM for External Users

Once the vTM EC2 instance is deployed, set up the instance using the Initial Configuration wizard. For details, refer Pulse Secure Virtual Traffic Manager: Cloud Services Installation and Getting Started Guide.
The Pulse Secure vTM Administrator login prompt appears.

Figure 45: Pulse Secure vTM Login Page

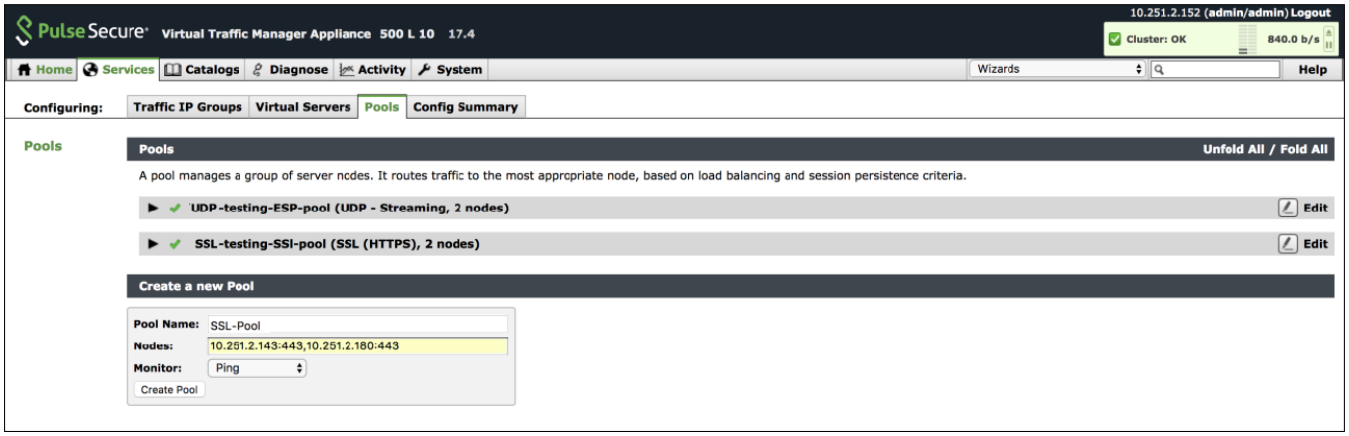


Next step is to set up the vTM for the external users using traffic pools and load balancing virtual servers. Traffic pool is the group that will bind to virtual server for load balancing. In an Active-Active Cluster scenario, traffic pool comprises cluster nodes. We need to create two separate traffic pools, each for SSL(L7) and ESP(L3) traffic modes.

Create Service Pool

In the **Services** tab, select **Pools** and create new pool by adding external IPs of cluster nodes along with port number. Also, select appropriate monitor from the drop-down options.
Complete these steps for SSL and UDP. For details, refer to the section “Creating PCS Pools” in Load Balancing PCS with vTM Deployment Guide.

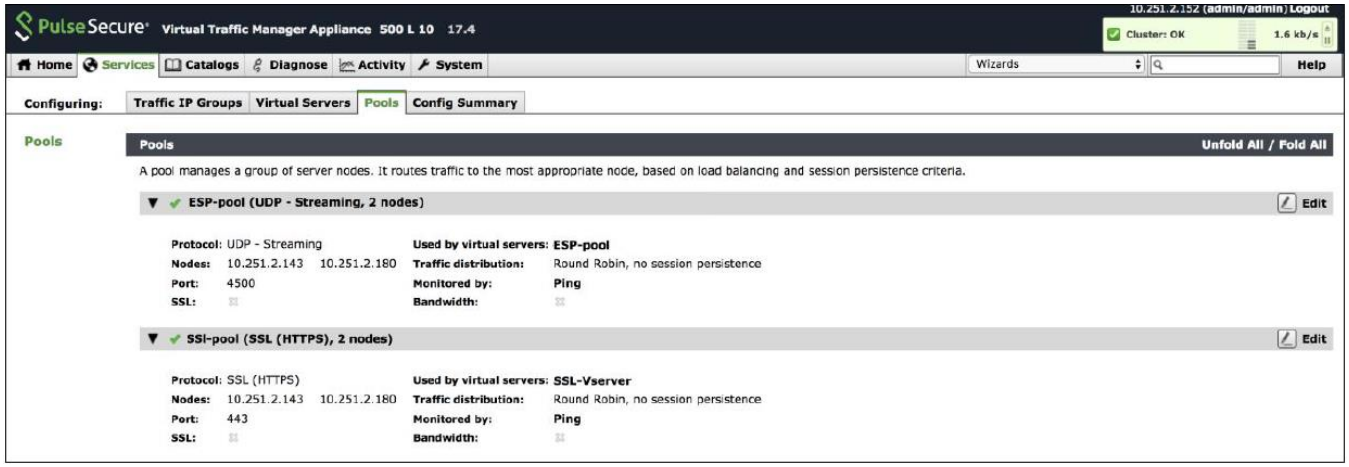
Figure 46: Create Traffic Pool



© 2019 Pulse Secure, LLC. All rights reserved

By default, they use Round Robin method of traffic distribution without any session persistency. Make a note of protocol type and port numbers that has been used for this use case.

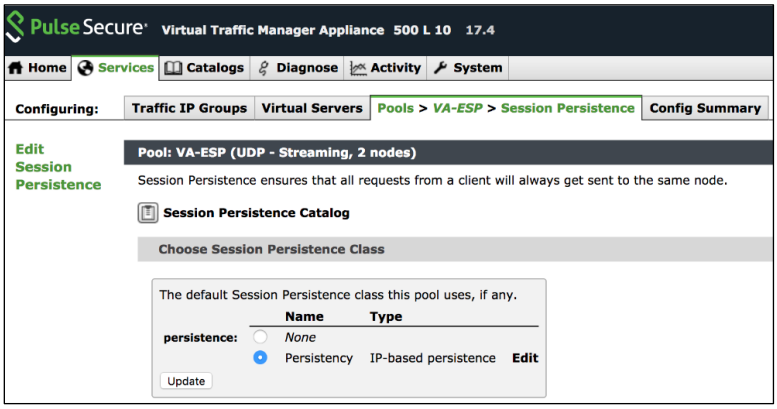
Figure 47: SSL and UDP Pools



Choose an IP-based Session Persistence Class

In the **Services** tab, select **Pools.** In the pool edit page, locate the Session Persistence section and enable the **Session Persistence** class. Session persistency is required for ESP-based VPN tunnels.

Figure 48: Session Persistency Class



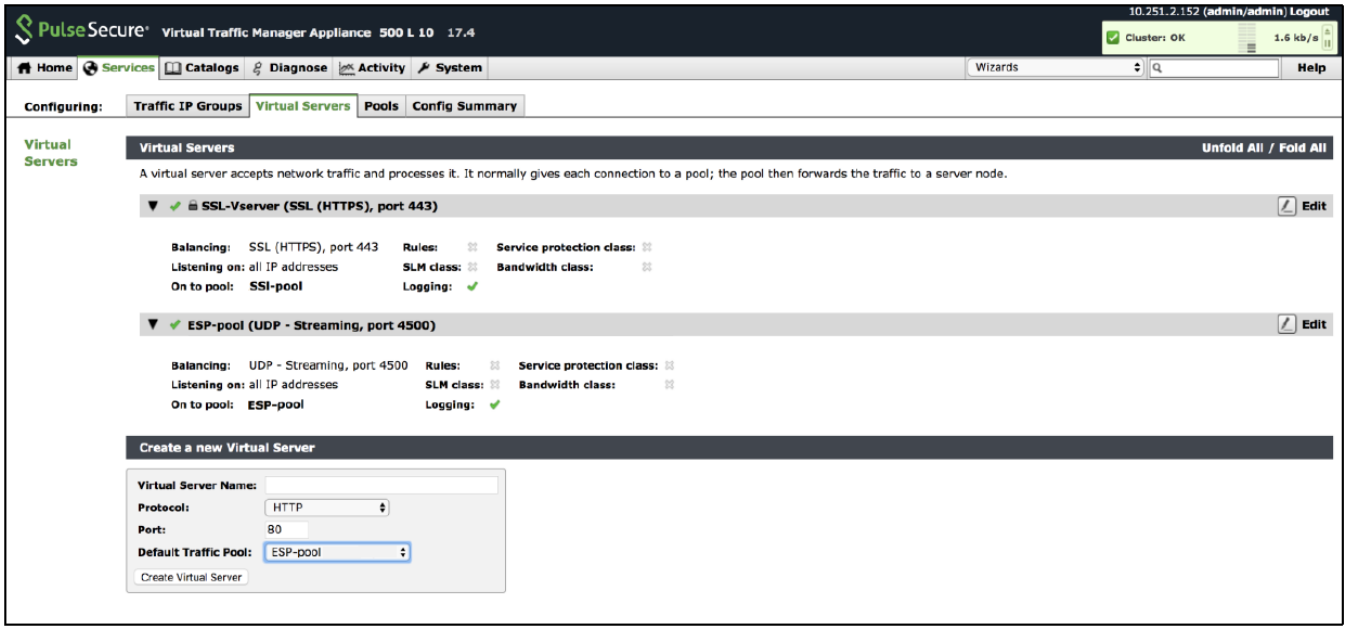
© 2019 Pulse Secure, LLC. All rights reserved

Create Virtual Servers

In the **Services** tab, select **Virtual Servers** and create a new virtual server by selecting protocol type and traffic pools. You need to create separate virtual servers to handle both SSL and UDP traffic. Each virtual server balances traffic across the pool of the same protocol type.
For details, refer to the section "Creating Virtual Server" in Load Balancing PCS with vTM Deployment Guide.

Figure 49: Create Virtual Server

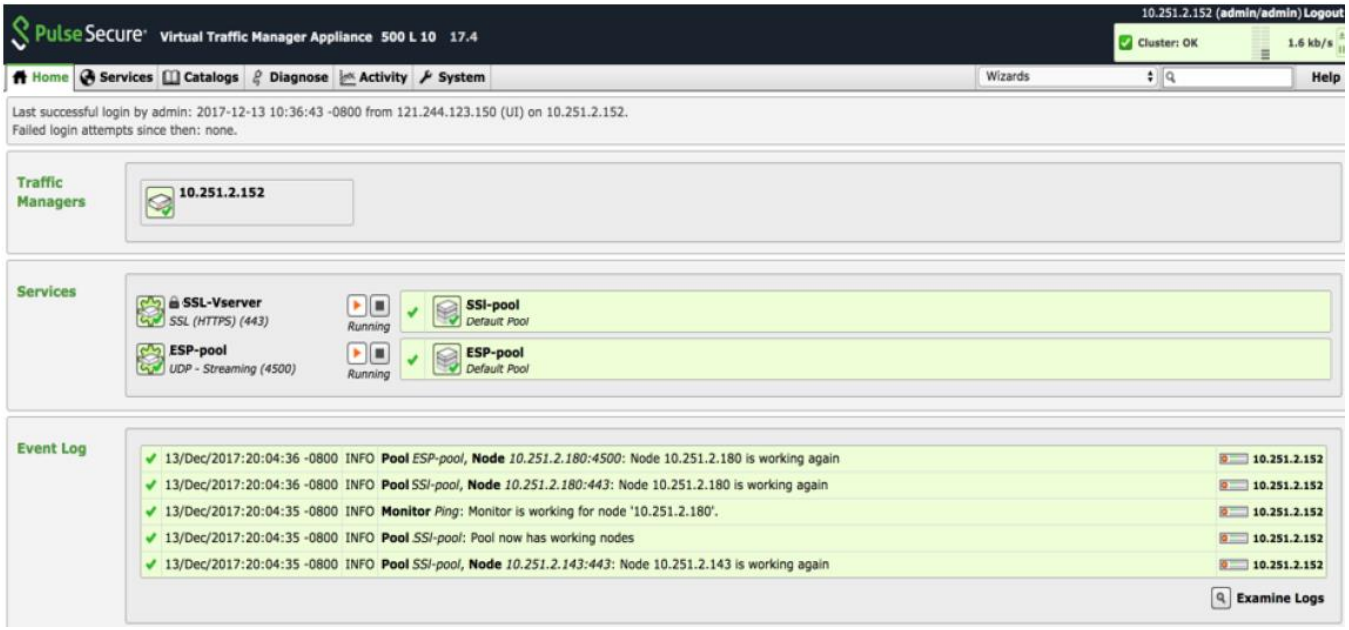
Figure 50: Virtual Servers to Handle SSL and UDP Traffic



© 2019 Pulse Secure, LLC. All rights reserved

Once the configuration is complete, go to home page and verify the configurations.

Figure 51: Pulse Secure vTM Home Page Showing Services and Event Logs



© 2019 Pulse Secure, LLC. All rights reserved

Accessing the Pulse Connect Secure Virtual Appliance

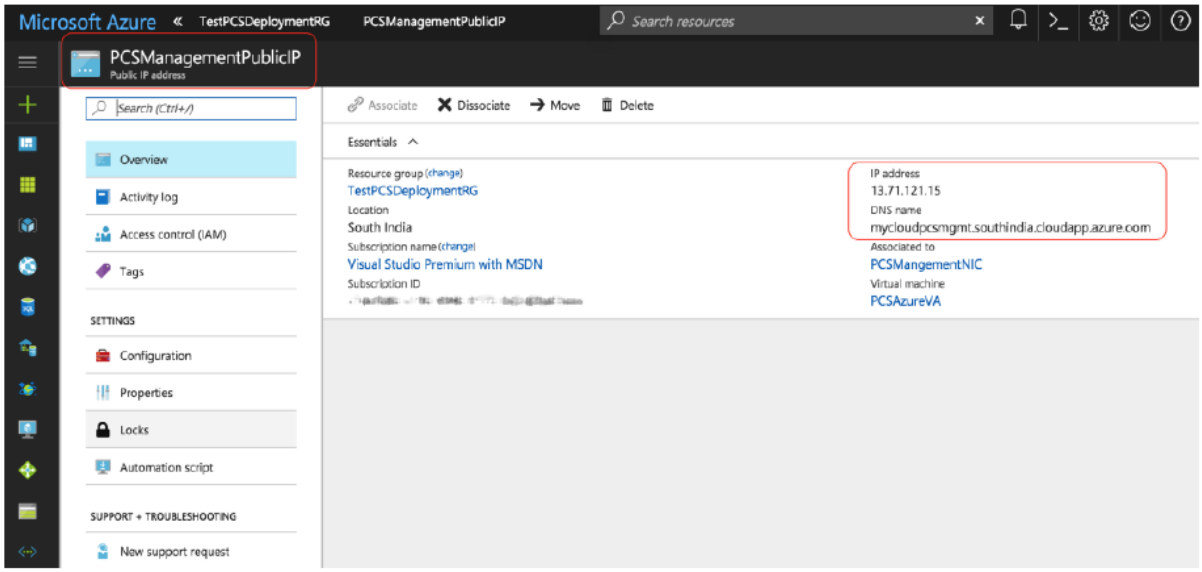
The Pulse Connect Secure appliance can be accessed:

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

Accessing the Pulse Connect Secure Virtual Appliance as an Administrator

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

Figure 52: Pulse Management Interface



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

The administrator can configure Active Directory located in the corporate network for user authentication. The Pulse Connect Secure Virtual Appliance administrator can check troubleshooting tools provided in the Pulse Connect Secure admin UI (System->Maintenance->Troubleshooting), to verify whether Pulse Connect Secure is able to reach other cloud resources as well as corporate resources. For this, Azure network administrator needs to ensure that all other resources have Pulse Connect Secure Internal interface as its default gateway.

© 2019 Pulse Secure, LLC. All rights reserved

Accessing the Pulse Connect Secure Virtual Appliance as an End User

To access the Pulse Connect Secure Virtual Appliance as an end user, copy the IP address from Pulse External Interface resource.

Figure 53: Pulse External Interface

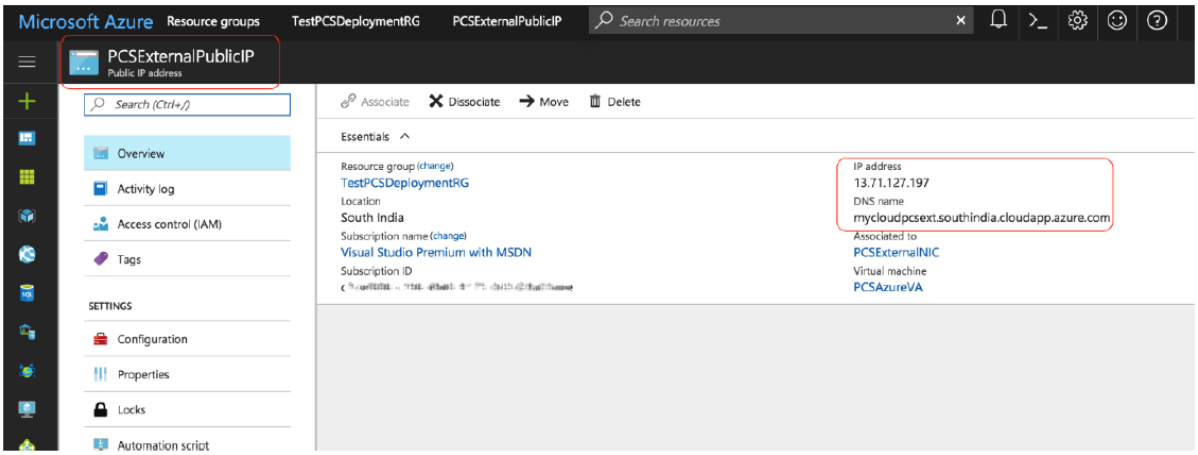
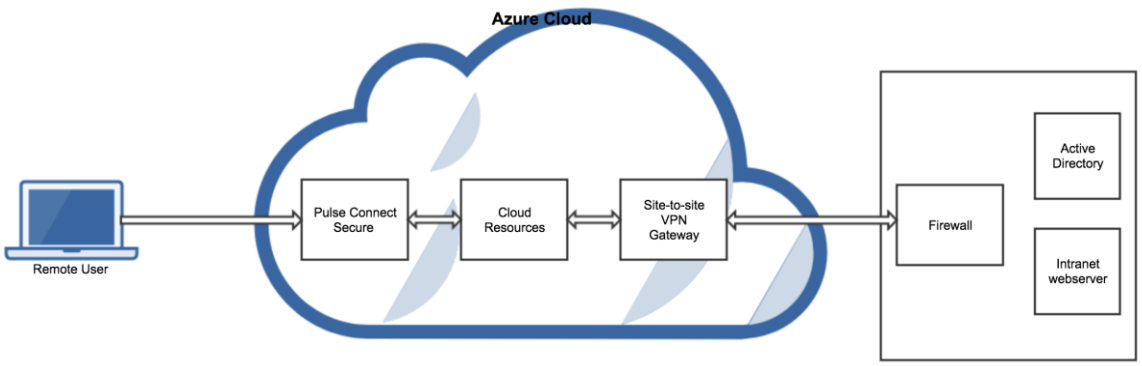


Figure 54: Resource in Corporate Network



Accessing the Pulse Connect Secure Virtual Appliance using SSH Console

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

On Linux and Mac OSX

Execute the following command:

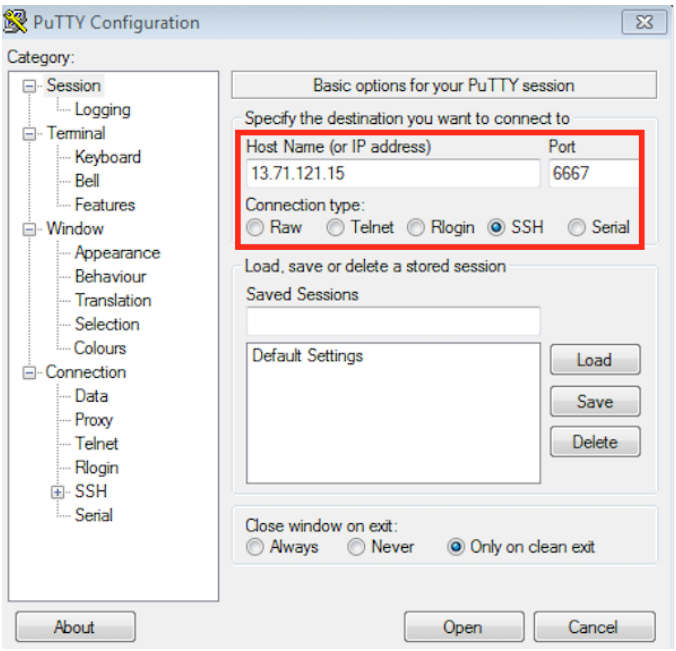
```
ssh -i <rsa-private-key-file> <PCS-Management-Interface-PublicIP> -p 6667
```

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.

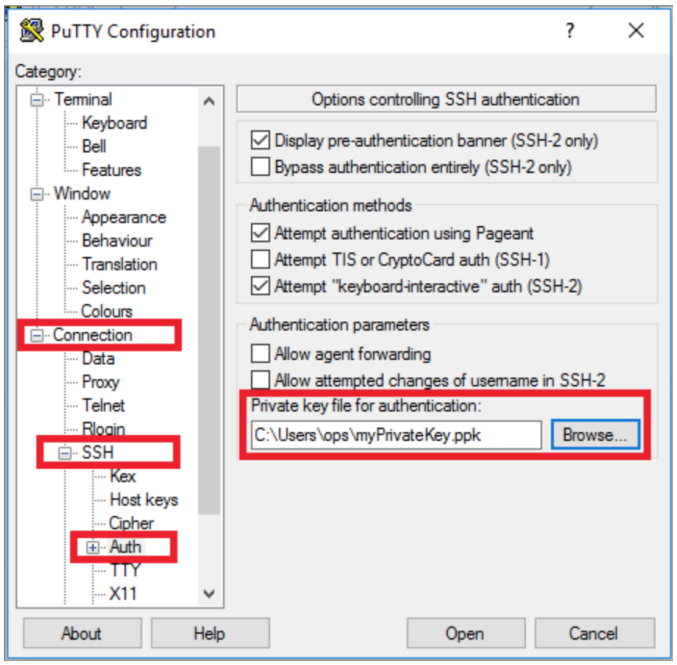
© 2019 Pulse Secure, LLC. All rights reserved

Figure 55: Putty Configuration – Basic Options



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

Figure 56: Putty Configuration – SSH Authentication

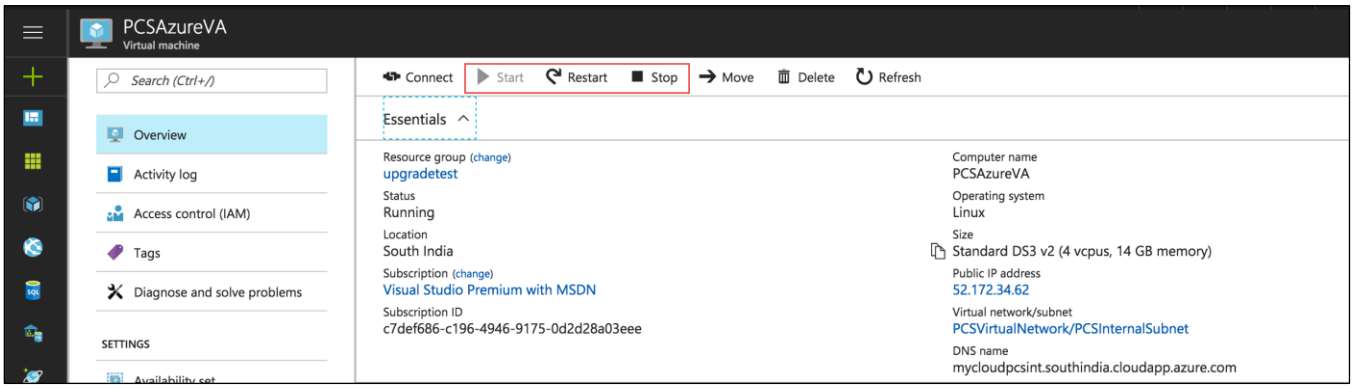


© 2019 Pulse Secure, LLC. All rights reserved

System Operations

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

Figure 57: 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 Templates provided by Pulse Connect Secure creates the Pulse Connect Secure Virtual Appliance with public and private IP addresses for external and management interfaces and only private IP address for internal 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 Connect Secure gets deployed by using the sample templates provided by Pulse Secure, Pulse Connect Secure comes up with multiple interfaces. If you take an example of a template “pulsesecure-pcs-3-nics.zip” provided by Pulse Secure, you notice the following things.

© 2019 Pulse Secure, LLC. All rights reserved

PCS external interface and PCS management interface are 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.

```
1.      "type": "Microsoft.Network/networkInterfaces",
2.      "name": "[variables('pcsExtNic')]",
3.       ---------------
4.       ---------------
5.      "properties": {
6.       "privateIPAllocationMethod": "Dynamic",
7.      "privateIPAddressVersion": "IPv4",
8.      "publicIPAddress": {
9.      "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. When you are assigning static IP address, make sure that it is not in the reserved IP category.

```
1.   "ipConfigurations": [{
2.      "name": "ipconfig2",
3.      "properties": {
4.         "privateIPAllocationMethod": "Static",
5.         "privateIPAddressVersion": "IPv4",
6.         "privateIPAddress": "[variables('privateIPExternal')]",
7.      }
8.   }]
```

Modifying Network Parameters After Deployment

Since Networking Infrastructure is provided by Azure, a PCS 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 PCS 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	PCS 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.

© 2019 Pulse Secure, LLC. All rights reserved

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 Connect 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.

```
1.   "networkProfile": {
2.      "networkInterfaces": [{
3.         "id": "nic1",
4.         "properties": {
5.            "primary": true
6.         }
7.      }, {
8.         "id": "nic2",
9.         "properties": {
10.           "primary": false
11.        }
12.     }, {
13.        "id": "nic3",
14.        "properties": {
15.           "primary": false
16.        }
17.     }]
18.  },
```

PCS 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	PCS 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 PCS. As a best practice, always assign 'primary' to the first network interface which will become internal interface of PCS.

© 2019 Pulse Secure, LLC. All rights reserved

Backing up Configs and Archived Logs on Azure Storage

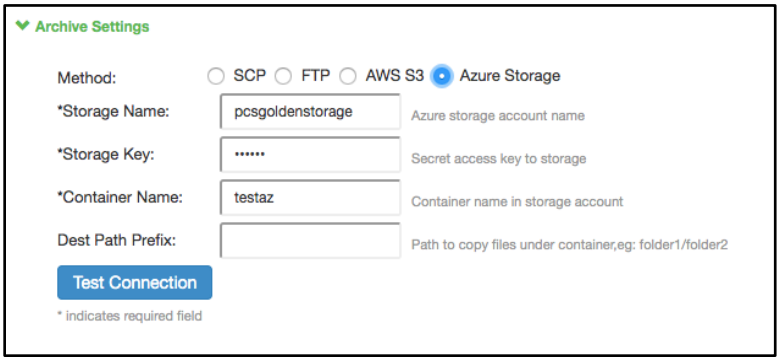
Pulse Connect Secure supports pushing configs and archived logs to the servers that support SCP and FTP protocols. In the Azure deployment, Pulse Connect Secure now supports pushing configs and archived logs to the Azure storage.

Configuring Backup Configs and Archived Logs via PCS Admin Console

To configure backing up configs and archived logs:

1. Log into the Pulse Connect 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 58: 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-

© 2019 Pulse Secure, LLC. All rights reserved

	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 Connect Secure

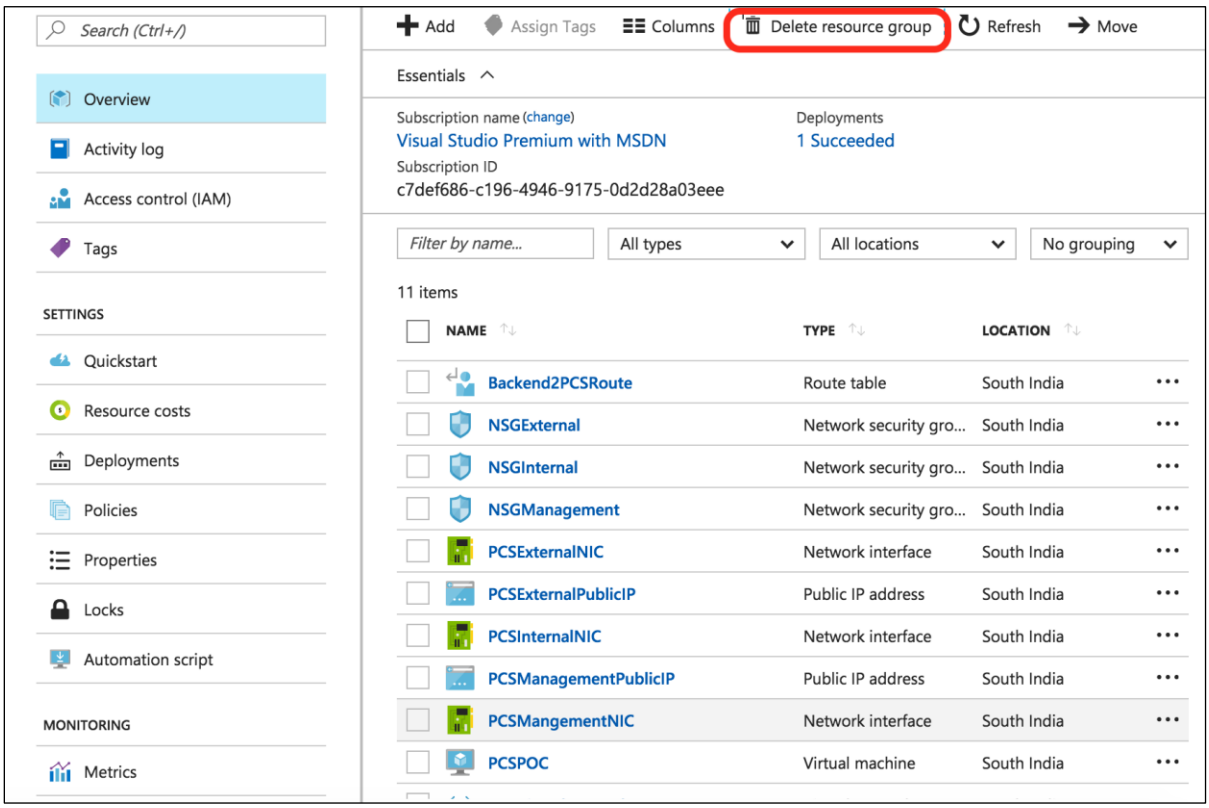
When deploying Pulse Connect Secure, if you have selected the option “Use existing resource group”, then follow the steps mentioned in the section Delete Pulse Connect Secure and Resource It Uses, but not the Other Resources in Resource Group. Else if you have selected the option “New resource group”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

Delete Entire Resource Group that the Pulse Connect Secure Is In

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

© 2019 Pulse Secure, LLC. All rights reserved

Figure 59: Delete Resource Group



5. Navigate to the storage account where the Pulse Connect 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-<pcs-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 "<pcs-name><13 digit unique string>pcsOSDisk.vhd".

Delete Pulse Connect 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 Connect Secure is in.
4. Delete the following resources:
   - PCS Virtual Machine
   - Virtual Network named PCSVirtualNetwork
   - PCSInternalNIC, PCSExternalNIC and PCSManagementNIC
   - PCSExternalPublicIP and PCSManagementIP
   - Three Network Security Groups named NSGInternal, NSGExternal and NSGManagement
   - User-defined Routing table named Backend2PCSRoute
5. Navigate to the storage account where the Pulse Connect 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-<pcs-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 "<pcs-name><13-digit unique string>pcsOSDisk.vhd".

© 2019 Pulse Secure, LLC. All rights reserved

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
PCS VM (Standard_DS3_V2)	Compute	Yes
Virtual Network with four subnets	Networking	No
Three NIC cards named PCSInternalNIC, PCSExternalNIC and PCSManagementNIC	Networking	No
Two static Public IPs name PCSExternalPublicIP and PCSManagementIP	Networking	Yes
Three Network Security Groups named NSGInternal, NSGExternal and NSGManagement.	Networking	No
User Defined Routing table named Backend2PCSRoute	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 "<pcs-name><13 digit unique string>pcsOSDisk.vhd"	Storage	Yes

Limitations

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

- VLAN tagging
- IPv6 capabilities
- Layer 3 Tunnel IP pool assignment via DHCP
- Layer 2 functionality like ARP Cache and ND Cache
- Virtual Ports
- Multicast capabilities
- Bandwidth management

Not Qualified

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

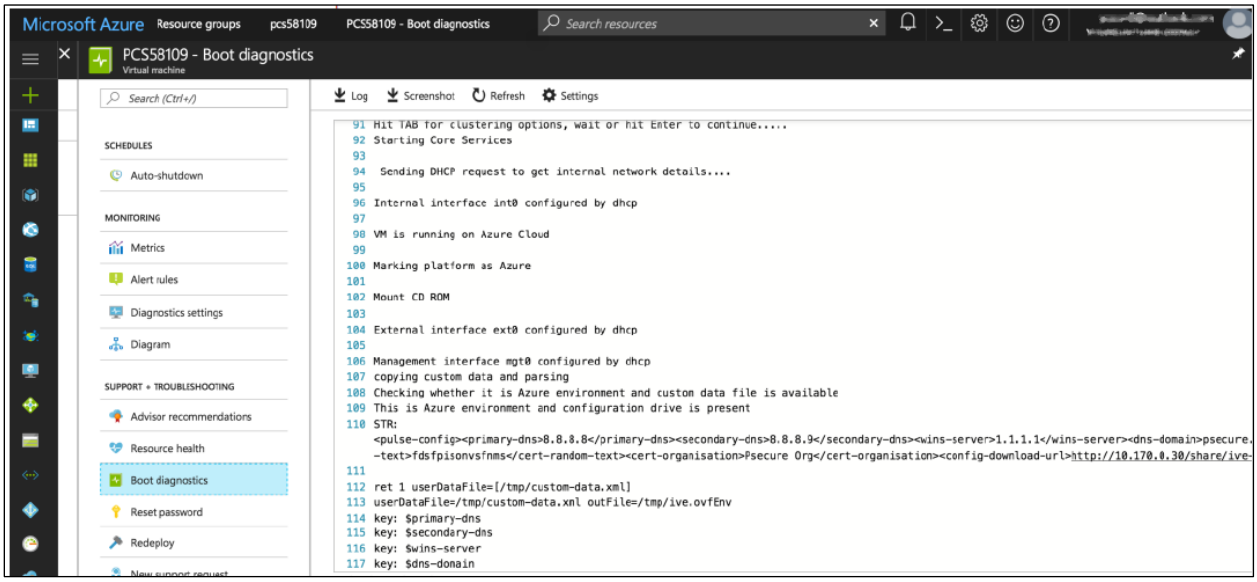
- Pulse Connect Secure and Pulse One interaction
- Pulse Connect Secure and PWS interaction
- IF-MAP support

© 2019 Pulse Secure, LLC. All rights reserved

Troubleshooting

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

Figure 60: Boot Diagnostics



© 2019 Pulse Secure, LLC. All rights reserved

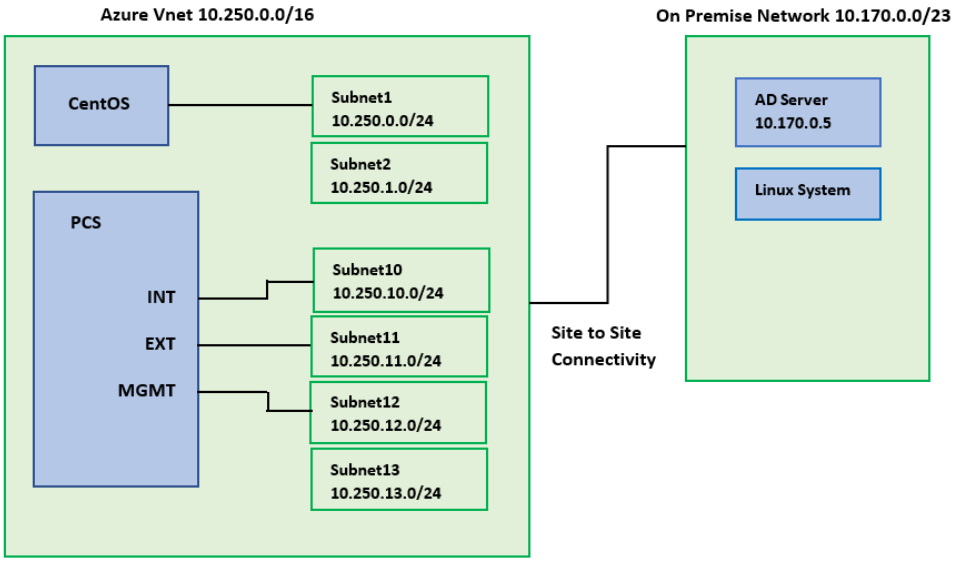
Frequently Asked Questions

This section provides solution for the frequently asked questions.

FAQ1: I am unable to connect to my backend resources through L3 VPN

Solution: The solution describes the configuration required in the Azure Virtual Network and PCS to connect to the On-premise network through the L3 VPN connection.
The following network topology shows two networks, Azure Virtual Network and On-premise network, with Site-to-Site connectivity between them.

Figure 61: Pulse Connect Secure on Microsoft Azure



Before proceeding with the configuration, prepare a configuration checklist that will be handy during the configuration.

Azure Virtual Network	Resource group name: OnPremRG Virtual Network name: VirtualNetwork Virtual Network address space: 10.250.0.0/16 Subnets: Subnet1: 10.250.0.0/24 Subnet2: 10.250.1.0/24 Subnet3: 10.250.2.0/24 Subnet4: 10.250.3.0/24 . . . . Subnet10: 10.250.10.0/24 Subnet11: 10.250.11.0/24 Subnet12: 10.250.12.0/24 Subnet13: 10.250.13.0/24 PCS Internal interface connected to Subnet10 (10.250.10.0/24) PCS External interface connected to Subnet11 (10.250.11.0/24) PCS Management interface connected to Subnet12 (10.250.12.0/24) PCS VPN pool connected to Subnet13 (10.250.13.0/24)

© 2019 Pulse Secure, LLC. All rights reserved

	CentOS system IP address: 10.250.0.4 connected to Subnet1 (10.250.0.0) Public IP address: 104.211.245.193 **VPN Pool address space**: 10.250.13.0/24
On-premise network	**On-premise Network address space**: 10.170.0.0/23 **AD Server IP address**: 10.170.0.5

The Azure Vnet with address space 10.250.0.0/16 has four subnets - Subnet10 to Subnet12 - connected to PCS's Internal, External and Management interfaces respectively, and Subnet13 connected to PCS VPN pool.
The CentOS system is connected to Subnet1.

Figure 62: Virtual Network in a Resource Group (OnPremRG)

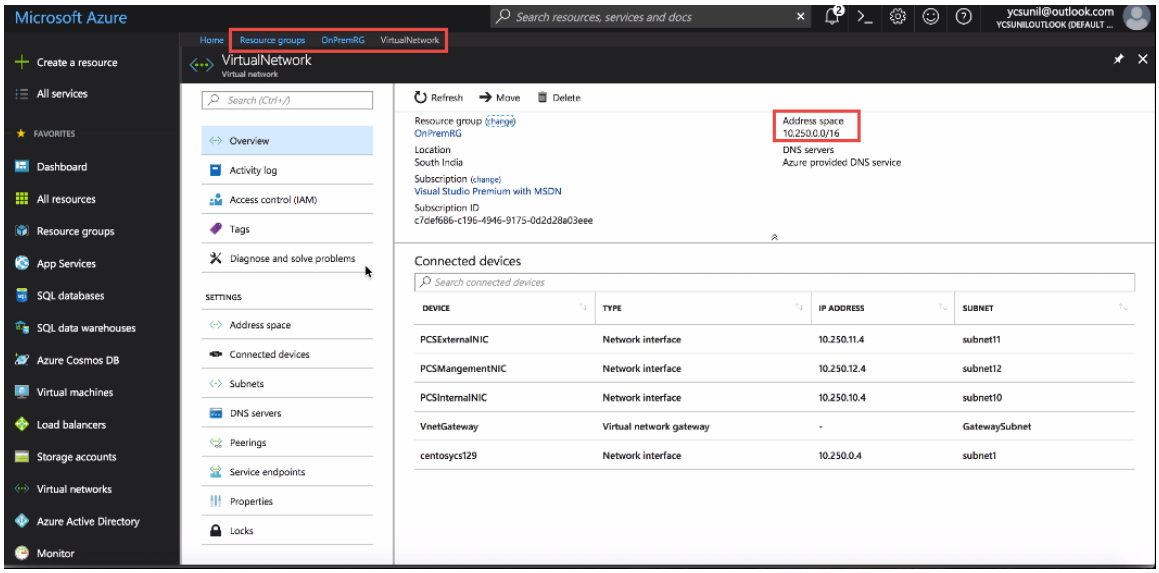
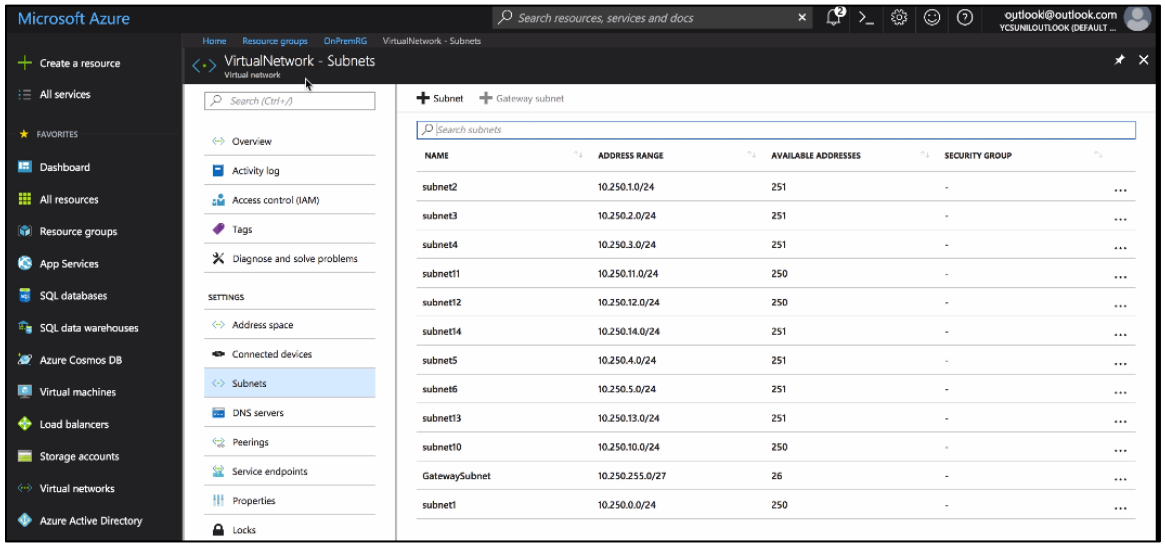


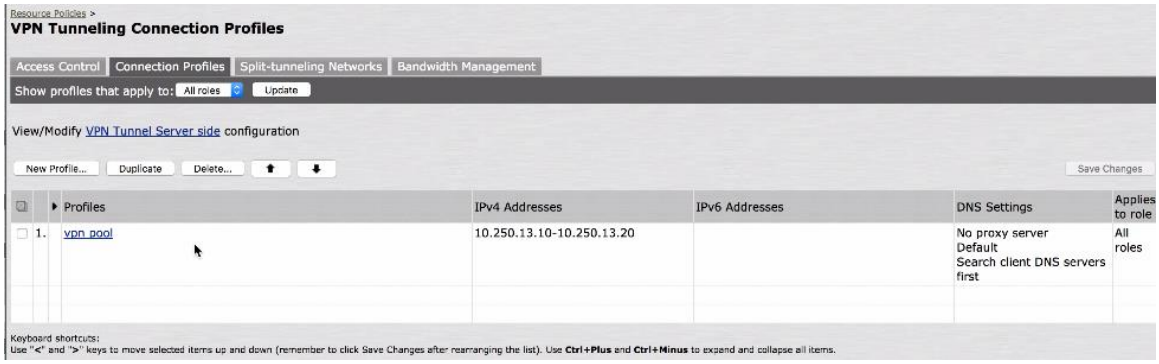
Figure 63: Subnets in the Virtual Network



© 2019 Pulse Secure, LLC. All rights reserved

Log in to PCS admin console and configure the VPN tunneling connection profile. The VPN pool has the range 10.250.13.10 to 10.250.13.20 in subnet13.

Figure 64: VPN Tunneling Connection Profile



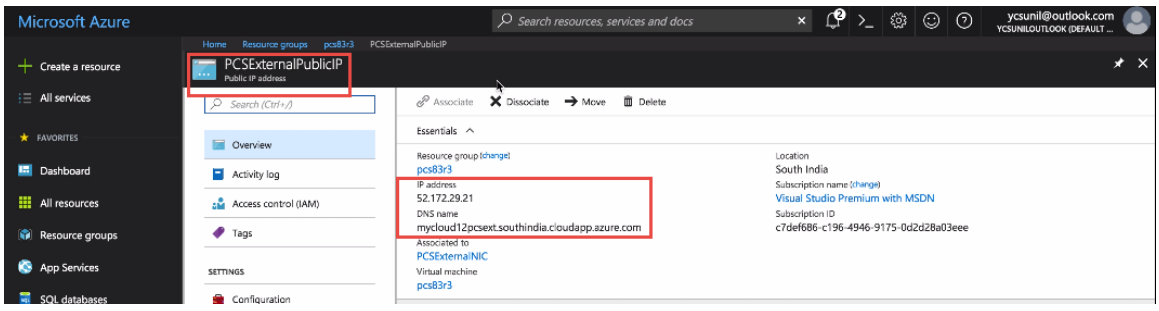
Create a user for this VPN tunnel policy and define the role mapping rule.

Figure 65: Use Role Mapping

Testing the Connection to CentOS System

1. Note down the public IP address / FQDN of PCS's External interface.

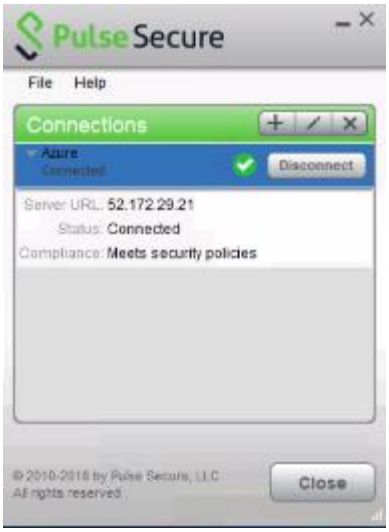
   Figure 66: Public IP of PCS External Interface



© 2019 Pulse Secure, LLC. All rights reserved

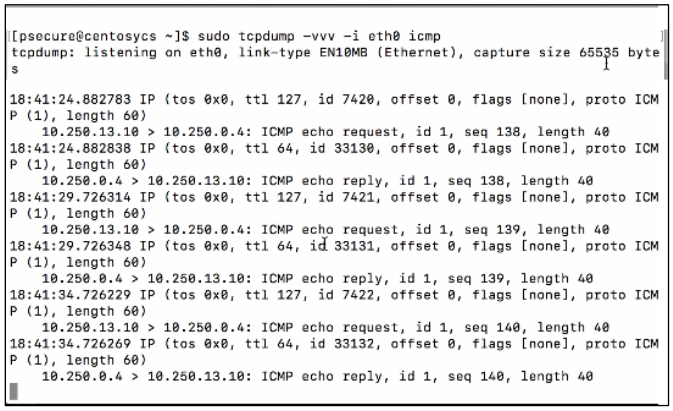
2. From client, connect to PCS.

Figure 67: Client Connection



3. Once connected, on the CentOS system run tcpdump to capture the icmp traffic. And from the client system, ping to CentOS system.

Figure 68: tcpdump and ping responses



The following is observed:

- The CentOS system shows echo request and echo reply messages.
- The ICMP request is from 10.250.13.10, which is the tunnel IP.
- The client system shows the “Request timed out” messages.
- The packet is sent out from the CentOS system, but it is not forwarded to the PCS Internal interface.

© 2019 Pulse Secure, LLC. All rights reserved

The solution is to add a route that forwards any packet in the tunnel IP address range 10.150.13.0/24 to PCS Internal interface. And associate the route to subnet (Subnet1) connected to the CentOS system.

Figure 69: Route Table

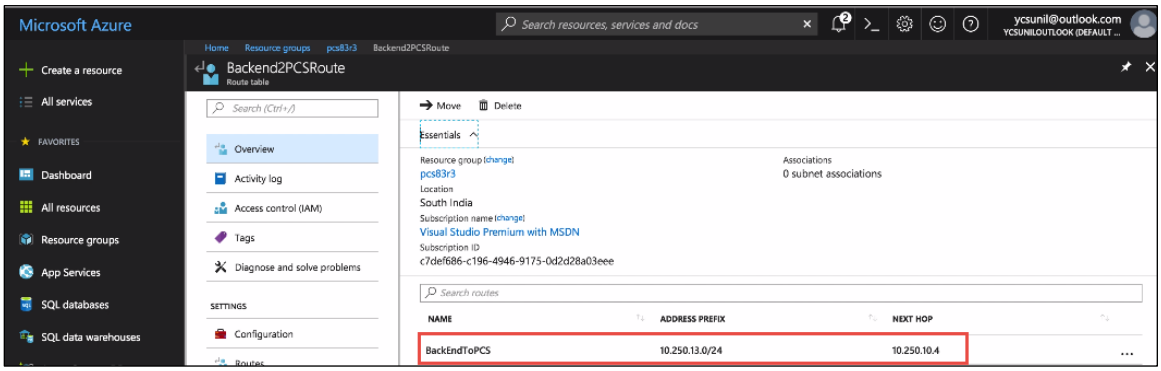
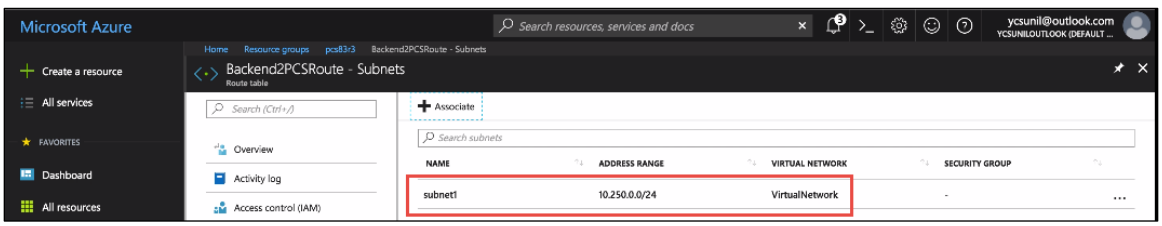
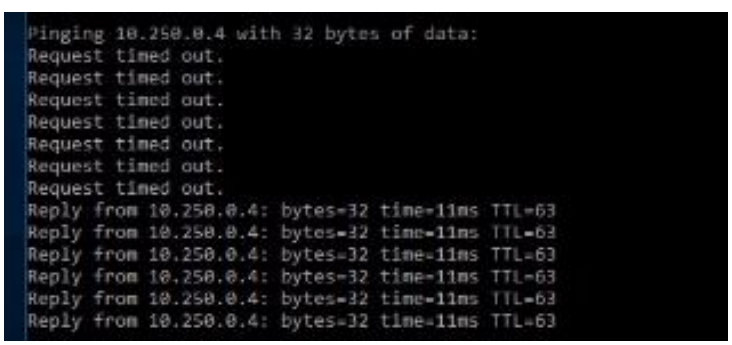


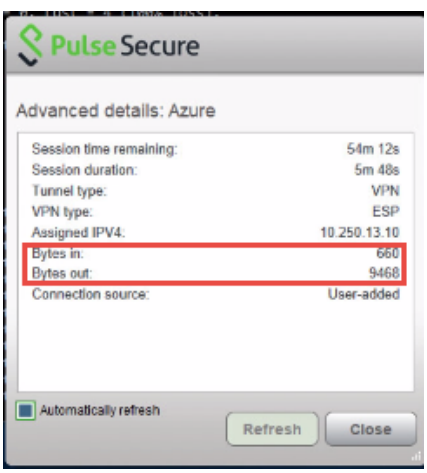
Figure 70: Subnet Association



Wait for some time and observe that the packets are transmitted successfully.

Figure 71: Successful Packets Transmission





© 2019 Pulse Secure, LLC. All rights reserved

Testing the Connection to On-premise Resource

1. From the client system, ping the on-premise resource, AD server whose IP address is 10.170.0.5.

```
C:\Users\admin>ping 10.170.0.5

Pinging 10.170.0.5 with 32 bytes of data:
Request timed out.
```

The output shows “Request timed out” messages. The packet in the return traffic stops at the Azure gateway subnet.
The solution is to add a route that forwards any packet at the gateway subnet to tunnel IP address range 10.150.13.0/24, and associate the route to gateway subnet.

Figure 72: Route Table

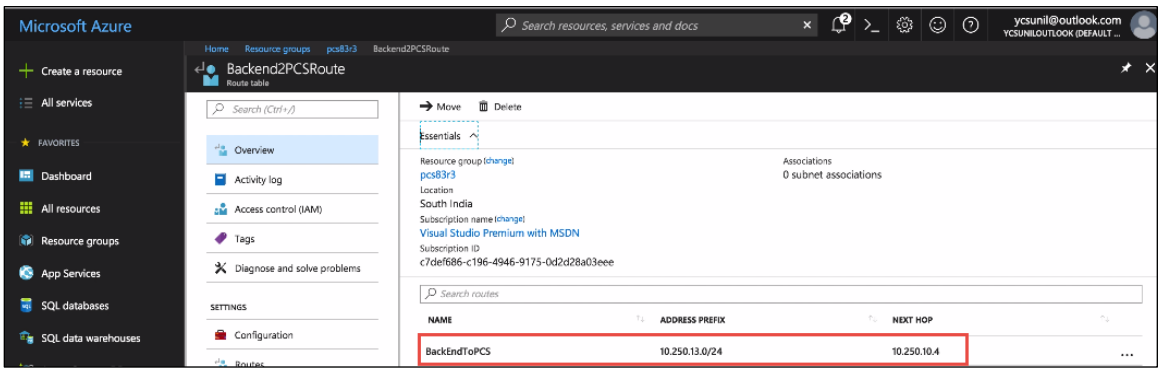
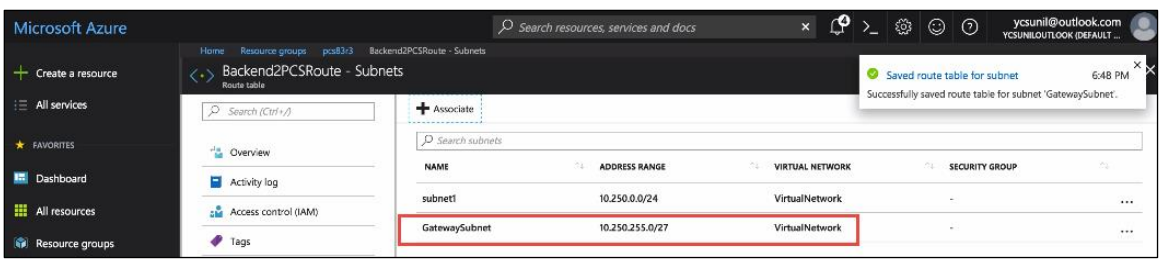
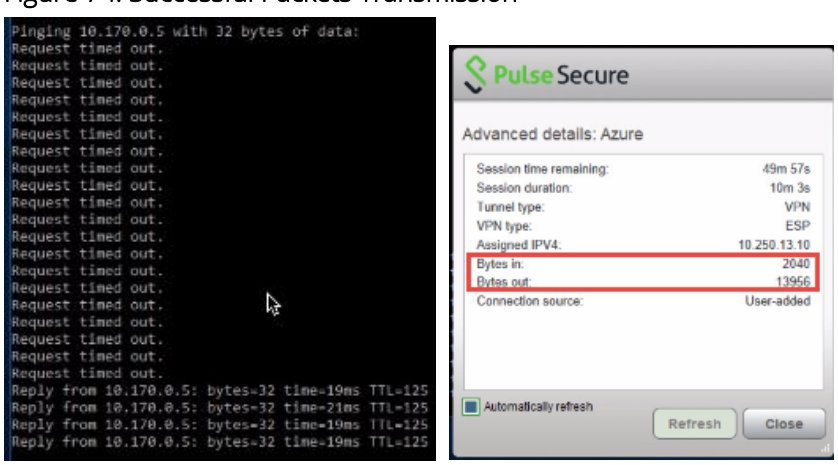


Figure 73: Subnet Association



Wait for some time and observe that the packets are transmitted successfully.

Figure 74: Successful Packets Transmission



© 2019 Pulse Secure, LLC. All rights reserved

FAQ2: Users are unable to access internet resources when connected to a VPN tunnel on an Azure-based PCS

Cause: When end user launches Pulse Client, connects to PCS in Azure and tries to access internet, PCS forwards the received packets (src ip: tunnel-ip, dest-ip: internet) through its internal interface. These packets reach Azure hidden Network Load Balancing (NLB). Azure hidden NLB drops these packets because it sees there is no NIC in the VNET with source IP as tunnel IP, the src-ip of the packet coming out of PCS is 'Client tunnel IP'.
Solution: Pulse Connect Secure must be able to SNAT these packets to the Internal interface IP which belongs to a subnet within the VNET.
To NAT endpoint tunnel IP to Internal interface IP, do the following:

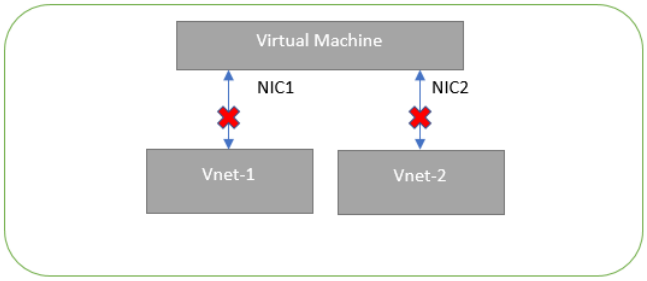
1. Log in to Pulse Connect Secure admin console.
2. Navigate to System > Network > VPN Tunneling.
3. Enable Source NATTING. By default, Source NATTING is disabled.

© 2019 Pulse Secure, LLC. All rights reserved

Appendix A: Network Security Group (NSG)

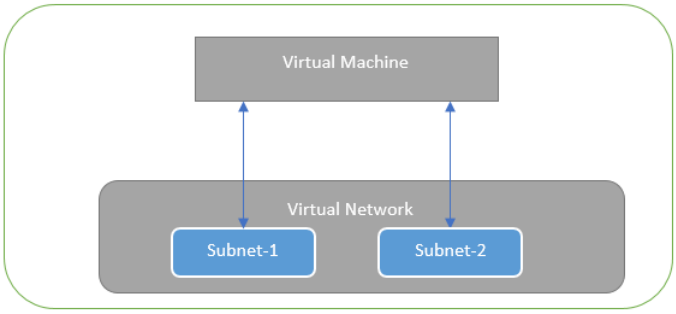
Microsoft Azure has a limitation where virtual machine with multiple network interfaces cannot connect to different Virtual Networks (VNETs). For example, a VM with two NIC cards, NIC1 and NIC2, will not be able to connect to Vnet1 and Vnet2 respectively.

Figure 75: Virtual Machine with two NIC cards Connecting to VNet1 and Vnet2



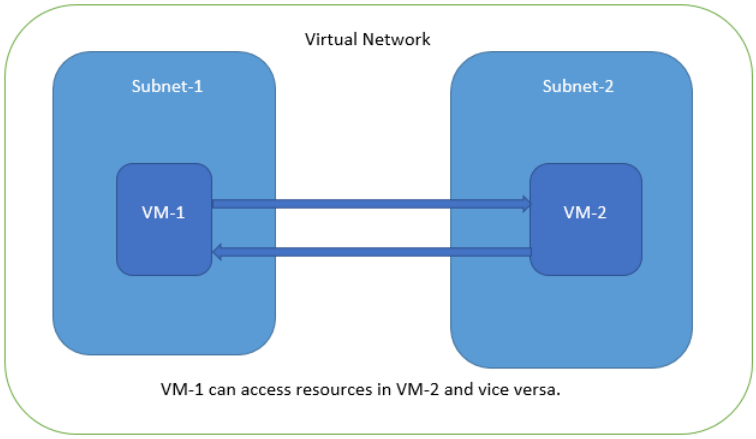
Microsoft Azure supports a virtual machine with multiple NICs to connect to different Subnets under a same Virtual Network. For example, a VM with two NICs, NIC1 and NIC2, can connect to 'Subnet1' and 'Subnet2' where these subnets exist under a same Virtual Network respectively.

Figure 76: 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.

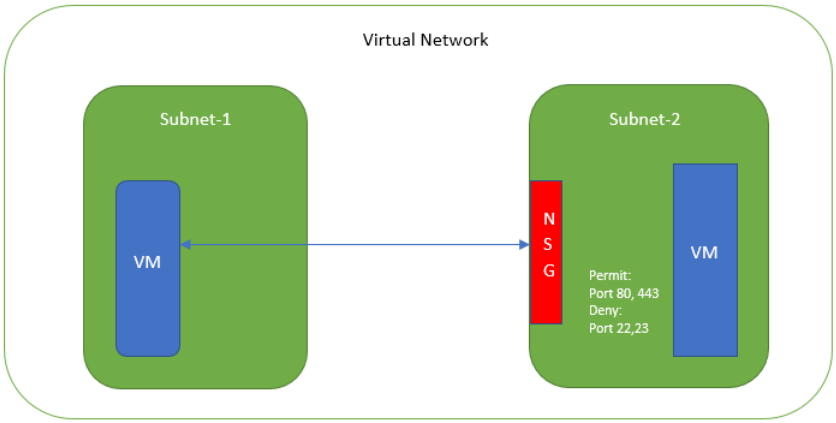
Figure 77: Virtual Machine VM-1 can Access Resources in VM-2 and Vice Versa



© 2019 Pulse Secure, LLC. All rights reserved

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 78: Traffic Filtering by MS Azure Network Support Group



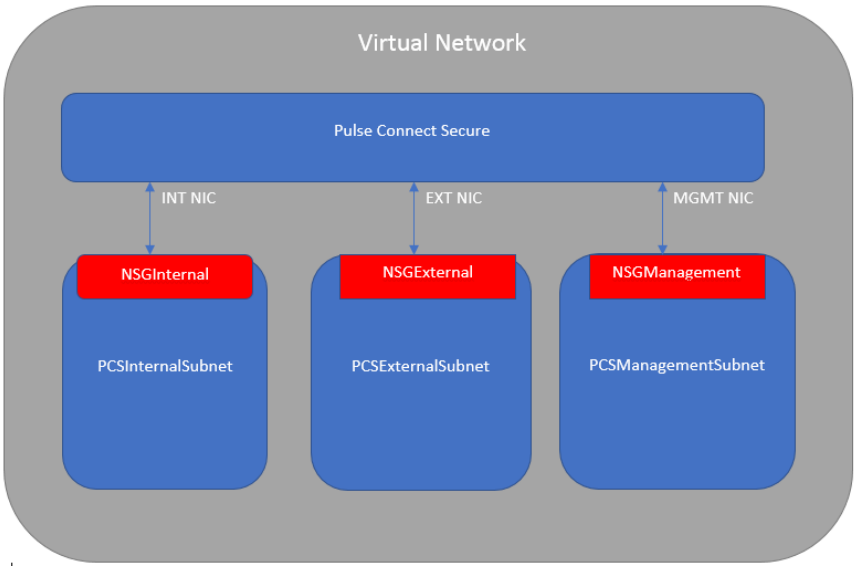
Pulse Connect Secure, when provisioned through the ARM template provided by Pulse Secure, creates four subnets under a virtual network named "PCSVirtualNetwork". The four Subnets are:

1. PCSInternalSubnet
2. PCSExternalSubnet
3. PCSManagementSubnet
4. PCSTunnelVPNPoolSubnet

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

1. NSGExternalSubnet
2. NSGInternalSubnet
3. NSGManagementSubnet

Figure 79: 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:

© 2019 Pulse Secure, LLC. All rights reserved

Figure 80: NSG External - Inbound Rules

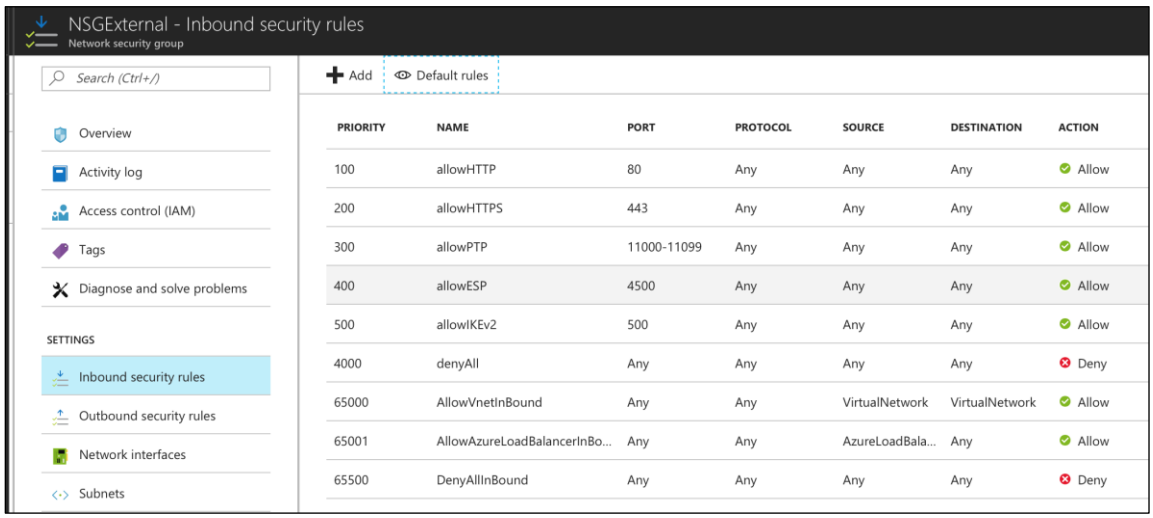
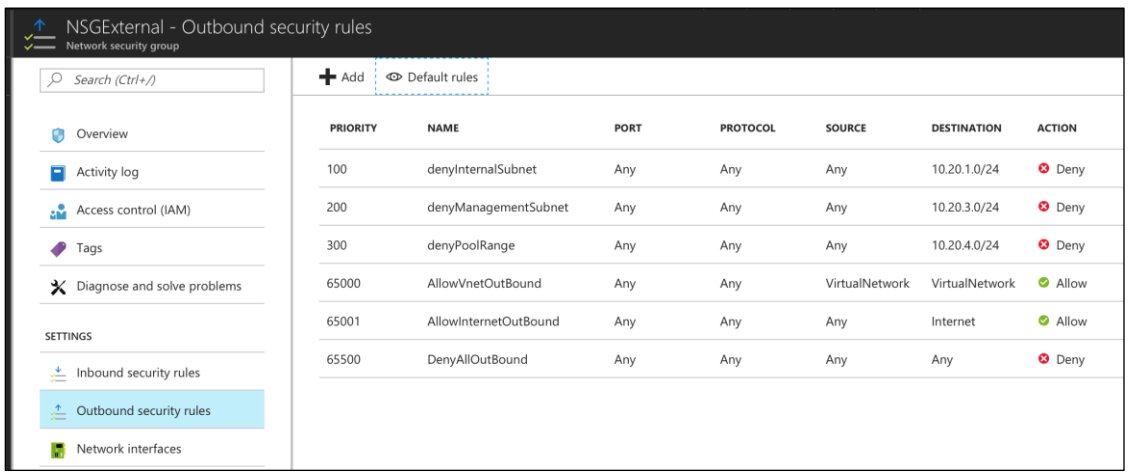
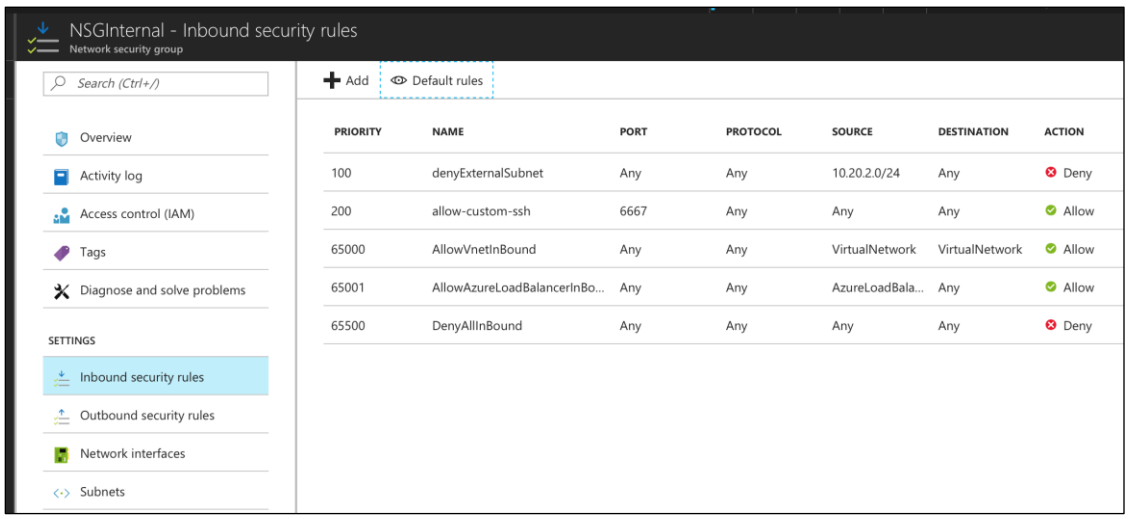


Figure 81: NSG External - Outbound Rules



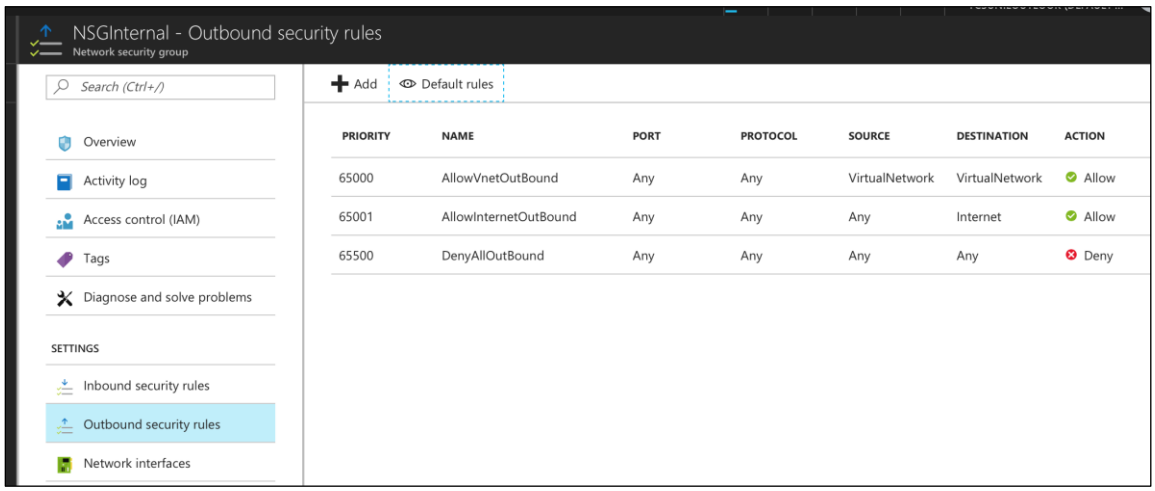
2. The list of NSG Inbound/Outbound rules created "**NSGInternalSubnet**" are:

Figure 82: NSG Internal - Inbound Rules



© 2019 Pulse Secure, LLC. All rights reserved

Figure 83: NSG Internal - Outbound Rules



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

Figure 84: NSG Management - Inbound Rules

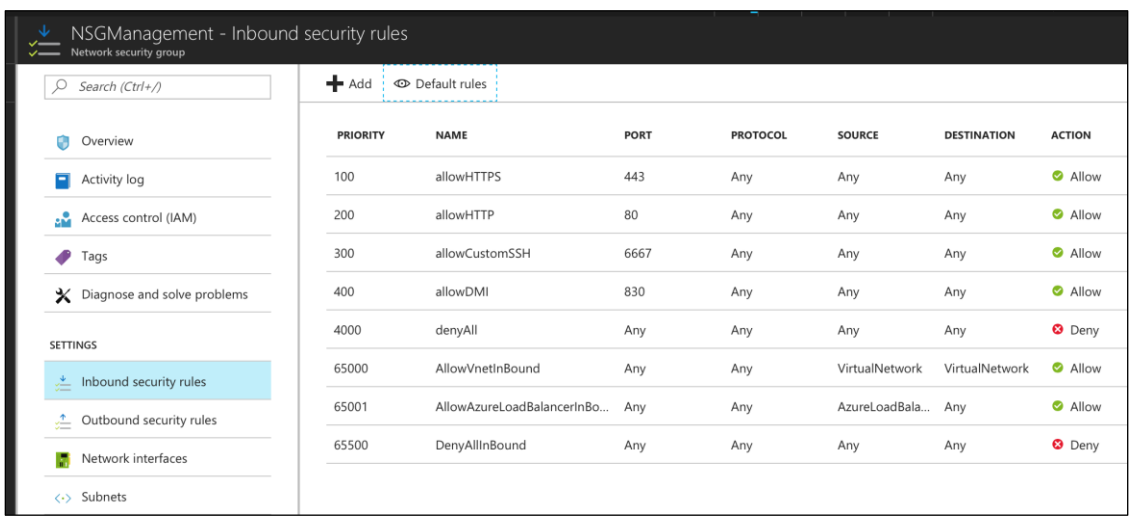
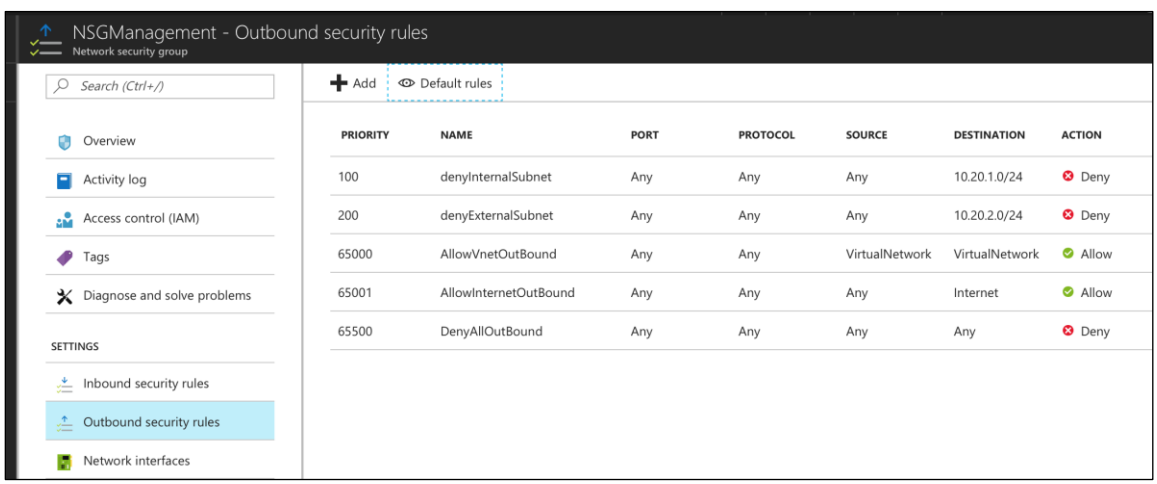


Figure 85: NSG Management - Outbound Rules



© 2019 Pulse Secure, LLC. All rights reserved

Appendix B: Pulse Connect Secure Resource Manager Template

Pulse Secure provides sample Azure template files to deploy the Pulse Connect 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-pcs-3-nics.zip file, and unzip it to get azuredeploy.json.

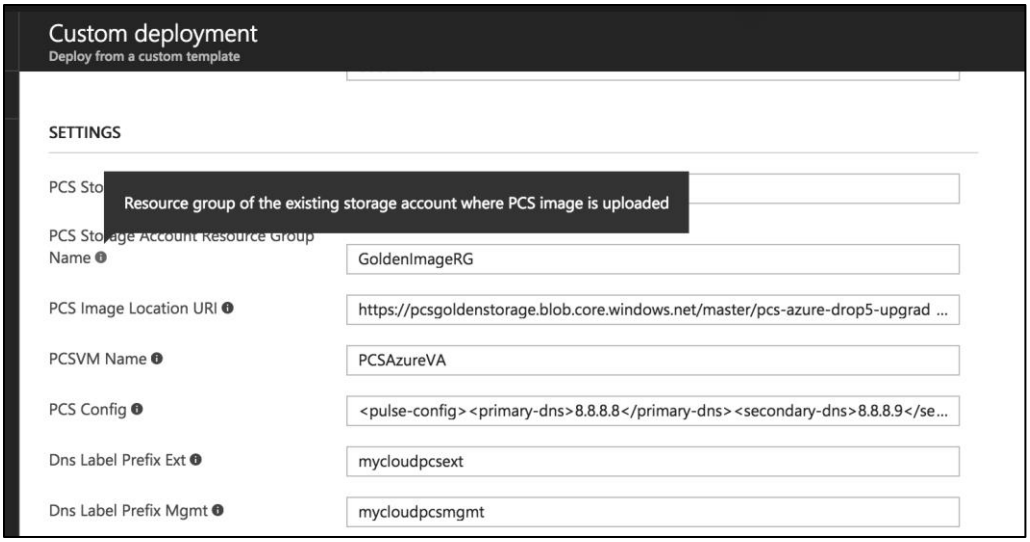
This template creates a new PCS with 3 NICs, Vnet, four subnets, NSG policies attached to PCS internal, external and management subnets and user-defined routes on the PCS internal subnet to ensure PCS is used as default gateway for L3 tunnel. All 3 NICs of PCS are configured with dynamic IP configuration and enabled IP forwarding. Public IPs are attached to the PCS external and management NIC.

The template has following sections:

parameters	This section defines the parameters used for deploying PCS 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 PCS on Azure.

parameters

Figure 86: Custom Deployment



© 2019 Pulse Secure, LLC. All rights reserved

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

```
"parameters": {
    "PCSStorageAccountName": {
    "type": "string",
    "defaultValue": "pcsgoldenstorage",
    "metadata": {
      "description": "Storage account name where PCS image is uploaded"
    }
```

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

```
"PCSStorageAccountResourceGroupName": {
  "type": "string",
  "defaultValue": "GoldenImageRG",
  "metadata": {
    "description": "Resource group of the existing storage account where PCS image is uploaded"
  }
},
```

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

```
"PCSImageLocationURI": {
  "type": "string",
   "defaultValue": "https://pcsgoldenstorage.blob.core.windows.net/master/pcs-azure-drop5-upgrade.vhd",
  "metadata": {
    "description": "URL of PCS vhd image"
  }
},
```

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

```
"PCSVMName": {
  "type": "string",
  "defaultValue": "PCSAzureVA",
  "metadata": {
  "description": "PCS VA Name"
        }
      },
```

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

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

PCS 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 Connect Secure through site-to-site VPN between Azure and the corporate data center.

Pulse Connect 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

© 2019 Pulse Secure, LLC. All rights reserved

- 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 Connect 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 Ext: This is the prefix for External Interface DNS label.

```
"dnsLabelPrefixExt": {
  "type": "string",
 "defaultValue": "mycloudpcsext",
  "metadata": {
    "description": "Unique DNS Name for the Public IP used to access PCS"
  }
},
```

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

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

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

```
"VnetAddressSpace": {
    "type": "string",
    "defaultValue": "10.20.0.0/16",
    "metadata": {
        "description": "Virtual Network Address Space"
    }
},
```

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

```
"InternalSubnet": {
    "type": "string",
    "defaultValue": "10.20.1.0/24",
    "metadata": {
        "description": "PCS internal interface connects to this subnet"
    }
},
```

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

```
"ExternalSubnet": {
    "type": "string",
    "defaultValue": "10.20.2.0/24",
    "metadata": {
        "description": "PCS external interface connects to this subnet"
    }
},
```

© 2019 Pulse Secure, LLC. All rights reserved

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

```
"ManagementSubnet": {
    "type": "string",
    "defaultValue": "10.20.3.0/24",
    "metadata": {
        "description": "PCS management interface connects to this subnet"
    }
},
```

Tunnel Subnet: Subnet which will be configured as Tunnel IP pool in Pulse Connect Secure VPN profile.

```
"TunnelSubnet": {
    "type": "string",
    "defaultValue": "10.20.4.0/24",
    "metadata": {
        "description": "Subnet used for VPN Pools"
    }
}
```

variables

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

```
"pcsvnetname"                   : "PCSVirtualNetwork",
```

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

```
"pcsVnetIntSubnet"              : "PCSInternalSubnet",
```

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

```
"pcsVnetExtSubnet"              : "PCSExternalSubnet",
```

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

```
"pcsVnetMgmtSubnet"             : "PCSManagemnetSubnet",
```

PCS Tunnel VPN Pools Subnet: This is the variable associated with the Subnet which will be configured as Tunnel IP pool in Pulse Connect Secure VPN Profile.

```
"pcsVnetTunnelPool"             : "PCSTunnelVPNPoolSubnet",
```

Backend to PCS Route: This creates route table for accessing the backend resources in Pulse Connect Secure Internal Interface.

```
"routeTableName"                : "Backend2PCSRoute",
```

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

```
"pcsIntPrivateIP"               : "10.20.1.4",
```

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

```
"pcsIntNic"                     : "PCSInternalNIC",
```

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

```
"pcsExtNic"                     : "PCSExternalNIC",
```

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

```
"pcsMgmtNic"                    : "PCSMangementNIC",
```

© 2019 Pulse Secure, LLC. All rights reserved

PCS External Public IP: This is public IP address assigned to PCS External Subnet.

```
"publicIPAddr1"                 : "PCSExternalPublicIP",
```

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

```
"publicIPAddr2"                 : "PCSManagementPublicIP",
```

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 Name: This variable defines PCS Virtual Machine name.

```
"vmName"                        : "MyPCSVM",
```

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

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

Virtual Network ID: This variable defines PCS 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 external interface IP address of PCS.

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

publicIPAddresses/publicIPAddr2: This block is responsible for creating public IP address which is static in nature. This is used for management interface IP address of PCS.

```
"type": "Microsoft.Network/publicIPAddresses",
"name": "[variables('publicIPAddr2')]",
```

virtualNetworks/pcvnetname: This block is responsible for creating PCS Virtual Network name. The creation of

© 2019 Pulse Secure, LLC. All rights reserved

PCS Virtual Network name depends on:

- Backend to PCS route
- NSG Internal Subnet
- NSG External Subnet
- NSG Management Subnet

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

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

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

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

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

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

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

virtualNetworks/pcsVnetTunnelPool: This block is responsible for creating tunnel pool. The created tunnel pool is applied to PCS Tunnel Pool.

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

routeTables/routeTableName: This block is responsible for creating route table. The created route table is used for accessing the backend resources in PCS Internal interface.

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

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

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

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

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

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

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

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

© 2019 Pulse Secure, LLC. All rights reserved

- PCS Virtual Network name

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

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

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

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

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
- allowPTP
- allowESP
- allowIKEv2
- denyAll
- denyInternalSubnet
- denyManagementSubnet
- denyPoolRange

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
- denyInternalSubnet
- denyExternalSubnet

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')]",
```

© 2019 Pulse Secure, LLC. All rights reserved

The following security rules can be defined:

- denyExternalSubnet
- allow-custom-ssh

outputs

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

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

© 2019 Pulse Secure, LLC. All rights reserved

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

Pulse Secure provides sample Azure template files to deploy Pulse Connect 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-pcs-3-nics.zip file, and unzip it to get **azuredeploy.json**.

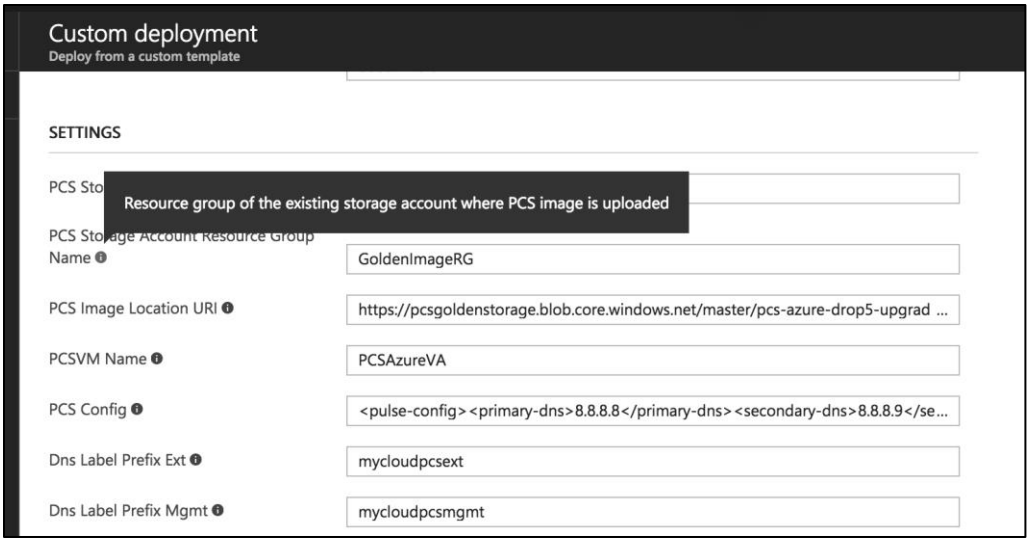
This template creates a new PCS with 3 NICs, Vnet, four subnets, NSG policies attached to PCS internal, external and management subnets and user-defined routes on the PCS internal subnet to ensure PCS is used as default gateway for L3 tunnel. All 3 NICs of PCS are configured with dynamic IP configuration and enabled IP forwarding. Public IPs are attached to the PCS external and management NIC.

The template has following sections:

parameters	This section defines the parameters used for deploying PCS 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 PCS on Azure.

parameters

Figure 87: Custom Deployment



© 2019 Pulse Secure, LLC. All rights reserved

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

```
"parameters": {
    "PCSStorageAccountName": {
    "type": "string",
    "defaultValue": "pcsgoldenstorage",
    "metadata": {
      "description": "Storage account name where PCS image is uploaded"
    }
  },
```

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

```
"PCSStorageAccountResourceGroupName": {
  "type": "string",
  "defaultValue": "GoldenImageRG",
  "metadata": {
    "description": "Resource group of the existing storage account where PCS image is uploaded"
  }
},
```

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

```
"PCSImageLocationURI": {
  "type": "string",
   "defaultValue": "https://pcsgoldenstorage.blob.core.windows.net/master/pcs-azure.vhd",
  "metadata": {
    "description": "URL of PCS vhd image"
  }
},
```

PCS VM Name: This is the name given to Pulse Connect Secure Virtual Appliance.

```
"PCSVMName": {
  "type": "string",
  "defaultValue": "PCSAzureVA",
  "metadata": {
  "description": "Pulse Connect Secure Name"
        }
      },
```

PCS 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 Connect Secure through site-to-site VPN between Azure and the corporate data center.
Pulse Connect 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

© 2019 Pulse Secure, LLC. All rights reserved

For details about these parameters, see Pulse Connect 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 Connect Secure via SSH.

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

DNS Label Prefix Ext: This is the prefix for External Interface DNS label.

```
"dnsLabelPrefixExt": {
  "type": "string",
 "defaultValue": "mycloudpcsext",
  "metadata": {
    "description": "Unique DNS Name for the Public IP used to access PCS"
  }
},
```

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

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

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 Connect Secure Internal Interface needs to lease IP.

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

© 2019 Pulse Secure, LLC. All rights reserved

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

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

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

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

Existing Tunnel Subnet: Subnet configured as Tunnel IP pool in Pulse Connect Secure VPN profile.

```
"existingTunnelSubnet": {
    "type": "string",
    "defaultValue": "subnet4",
    "metadata": {
        "description": "Subnet used for VPN Pools"
    }
}
```

variables

Backend to PCS Route: This creates route table for accessing the backend resources in Pulse Connect Secure Internal Interface.

```
"routeTableName"              : "Backend2PCSRoute",
```

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

```
"pcsIntNic"                   : "PCSInternalNIC",
```

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

```
"pcsExtNic"                   : "PCSExternalNIC",
```

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

```
"pcsMgmtNic"                  : "PCSMangementNIC",
```

PCS External Public IP: This is public IP address assigned to PCS External Subnet.

```
"publicIPAddr1"               : "PCSExternalPublicIP",
```

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

```
"publicIPAddr2"               : "PCSManagementPublicIP",
```

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",
```

© 2019 Pulse Secure, LLC. All rights reserved

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 PCS Virtual Machine size. It is 4 cores, 144MB memory.

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

Virtual Network ID: This variable defines PCS 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 external interface IP address of PCS.

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

publicIPAddresses/publicIPAddr2: This block is responsible for creating public IP address which is static in nature. This is used for management interface IP address of PCS.

```
"type": "Microsoft.Network/publicIPAddresses",
"name": "[variables('publicIPAddr2')]",
```

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
- allowPTP
- allowESP
- allowIKEv2
- denyAll
- denyInternalSubnet
- denyManagementSubnet

© 2019 Pulse Secure, LLC. All rights reserved

- denyPoolRange

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
- denyInternalSubnet
- denyExternalSubnet

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:

- denyExternalSubnet
- allow-custom-ssh

routeTables/routeTableName: This block is responsible for creating route table. The created route table is used for accessing the backend resources in PCS Internal interface.

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

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

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

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

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

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

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

networkInterfaces/pcsIntNic: This block is responsible for creating network interface. The created network interface is applied to network interface card of PCS Internal interface. The creation of this network interface

© 2019 Pulse Secure, LLC. All rights reserved

depends on:

- PCS Virtual Network name

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

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

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

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

outputs

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

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

© 2019 Pulse Secure, LLC. All rights reserved

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.

© 2019 Pulse Secure, LLC. All rights reserved