Pulse Connect Secure: Virtual Appliance on Amazon Web Services

Deployment Guide

Published September 2020

Document Version 5.1.1

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

www.pulsesecure.net

Pulse Connect Secure: Virtual Appliance on Amazon Web Services

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 http://www.pulsesecure.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Revision History

Revision and Date	Added/Updated/Removed	Remarks
5.1.1 September 2020	Updated the "Accessing the Pulse Connect Secure Virtual Appliance as an Administrator" section	Added PCS login instructions.
5.1 Jun 2020	Added Prerequisites section with SNAT configuration	
5.0 April 2020	Updated the “Pulse Connect Secure in AWS Marketplace” section.	Support for Terraform template
4.1 October 2019	Updated the “PCS Provisioning Parameters” section	
4.0 July 2019	Added backing up configs and archived logs on S3 bucket	Added the "Backing up Configs and Archived Logs on S3 Bucket" section
3.0 April 2019	Added Source NATTING	FAQ section is updated with the Source NATTING feature
2.0 March 2019	Added deploying on AWS Marketplace	Added Deploying 33PCS on AWS Marketplace
1.0, September 2018	None	Document has no changes from the previous release

Contents

Overview

About This Guide

This guide helps in deploying the Pulse Connect SecureVirtual Appliance on Amazon Web Services (AWS). In this release, Pulse Connect Secure is made available in AWS Market Place. A Pulse Connect Secure administrator can also manually upload the Pulse Connect Secure Virtual Appliance image (AMI) into AWS storage account. Once the AMI package is available in the AWS storage account, the Pulse Connect Secure administrator can deploy Pulse Connect Secure on AWS in the cloud.

Assumptions

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

Prerequisites

SNAT End Point Tunnel IP

The packets transmitted from PCS Internal Interface are dropped by AWS Virtual Gateway in L3 traffic. This is because the source IP and MAC address are not matching and the transit routing is not supported.

Pulse Connect Secure must be able to SNAT these packets to the Internal interface IP which belongs to a subnet within the VPC.

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.

Note: Enabling SNAT on PCS would reduce the number of connections, since one IP will be handling the traffic for all the end user Pulse client connections. So, it is recommended that you purchase a NAT gateway and assign it to PCS.

Pulse Connect Secure in AWS Marketplace

Beginning 9.0R4 release, Pulse Connect Secure is made available in AWS Market Place. The CloudFormation templates are available at Amazon marketplace.

Prerequisites and System Requirements on AWS Marketplace

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

- An AWS account
- Access to the AWS Marketplace (https://aws.amazon.com/marketplace)
- Pulse Connect Secure licenses *

Note:

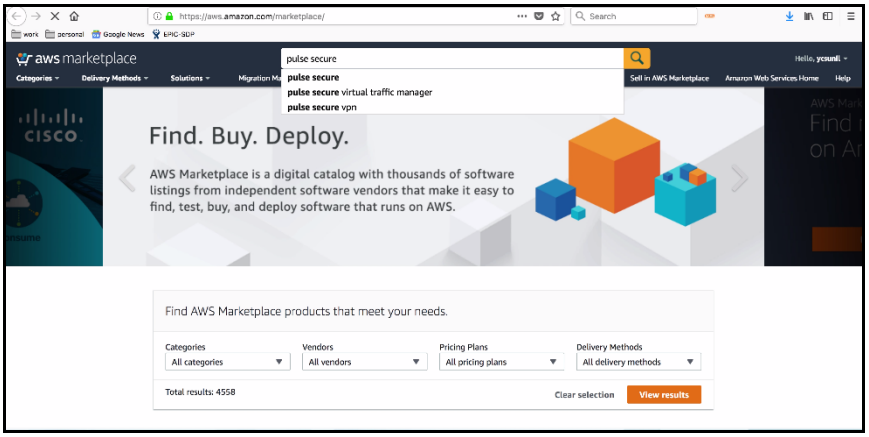
* From 9.0R3 release, Pulse Connect Secure Virtual Appliance, by default, has two evaluation licenses, and supports licensing with License server located at corporate network.

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

Deploying Pulse Connect Secure on AWS Marketplace

4. Launch AWS Marketplace using the URL: https://aws.amazon.com/marketplace and search with keyword Pulse Secure.

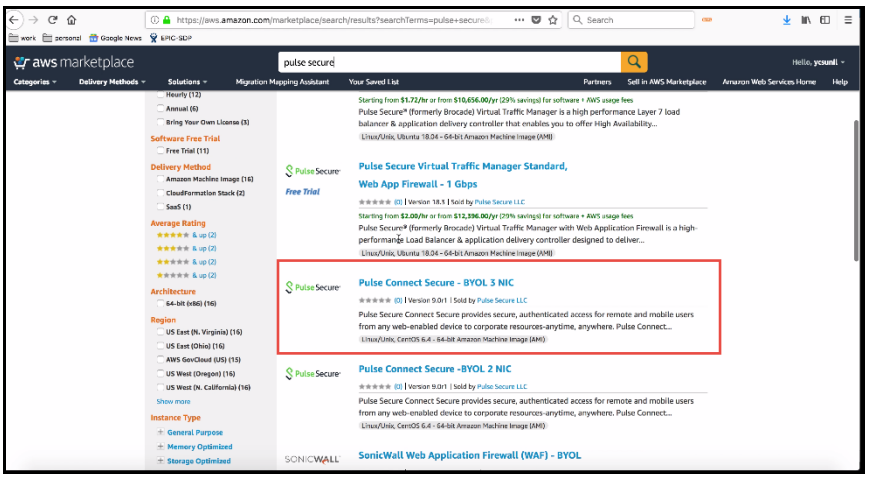
   Figure 1 AWS Marketplace

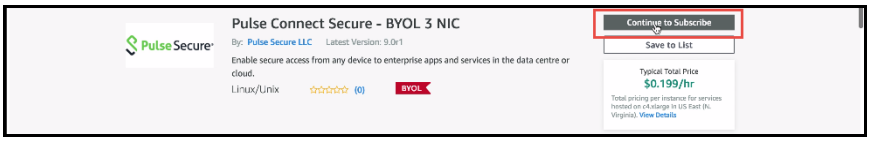


AWS Marketplace contains the following two Pulse Connect Secure SKUs:

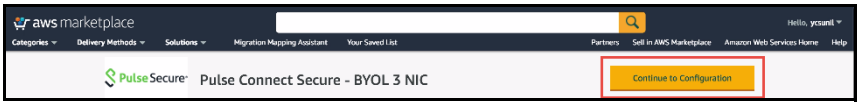
- Pulse Connect Secure - BYOL 2 NIC
- Pulse Connect Secure - BYOL 3 NIC

Figure 2 Subscribe to Pulse Connect Secure – BYOL 3 NIC



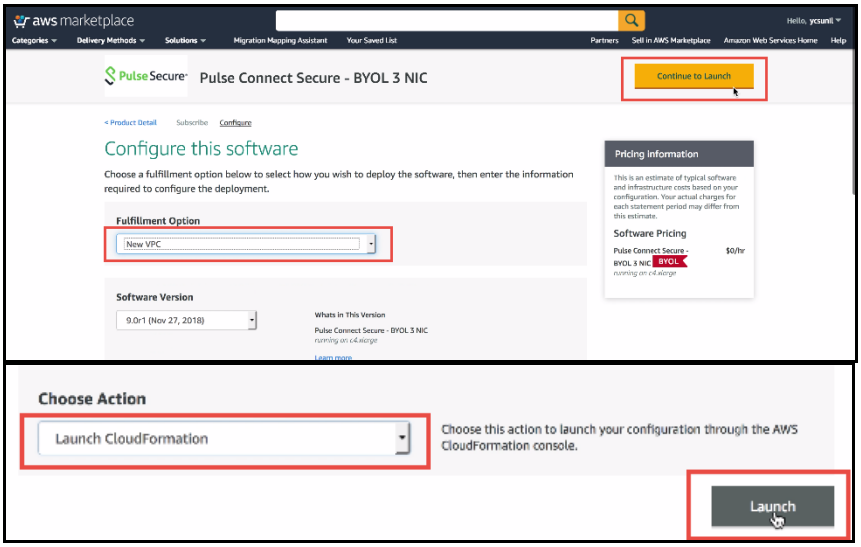


5. Select either 3-NIC model or 2-NIC model based on your requirement. In the Product Subscription page displayed, click **Continue to Subscribe**. In this section, 3-NIC model is chosen as example.
6. After subscribing, proceed to configuration by clicking **Continue to Configuration**.



7. In Fulfillment Option, select either Existing VPC or New VPC that you want to deploy and click **Continue to Launch**. In the Launch page displayed, select **Launch CloudFormation** and click **Launch**.

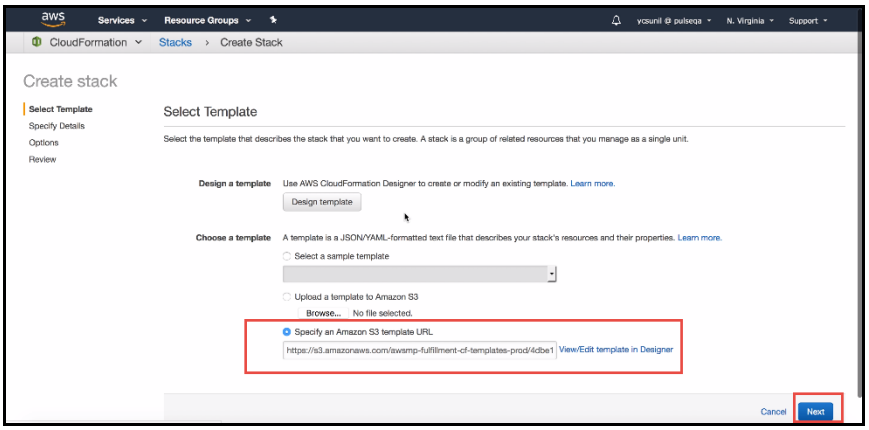
Figure 3 CloudFormation Template



Select Template

1. In the Create stack wizard, in the Select Template page choose the template that describes your stack's resources and their properties and, click **Next**.

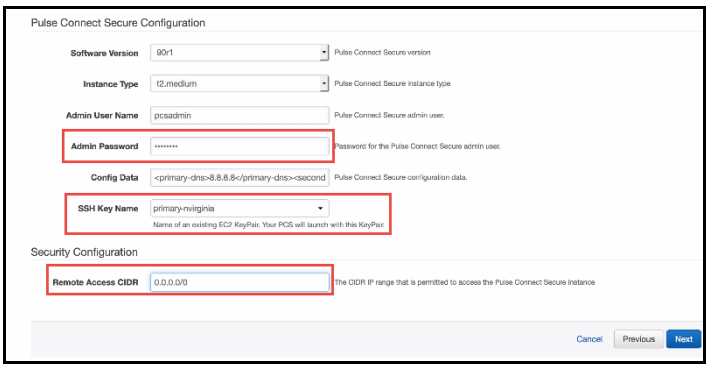
   Figure 4 Select Template



Specify Details

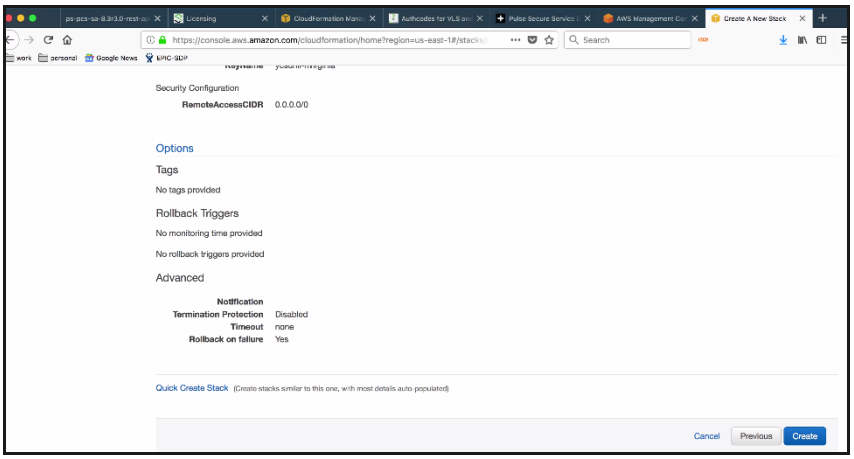
1. In the Specify Details page, specify a name for the stack.
2. In the Parameters section, use the default parameter values. These are defined in the CloudFormation template.
3. In the Pulse Connect Secure Configuration section:
   - Select Pulse Connect Secure VM size. By default it is set to t2.medium
   - By default, PCS admin user name is configured. You can give any other user name if you want to.
   - Enter the Admin user password.
   - Config Data: Provisioning parameters in an XML format. For details, see "Pulse Connect Secure Provisioning Parameters" on page 34.
   - Select SSH Key Name of EC2 key pair. 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 http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html.
   - In the Security Configuration section, enter Remote Access CIDR IP range that permits end user access to Pulse Connect Secure instance.

Figure 5 Specify Configuration Details

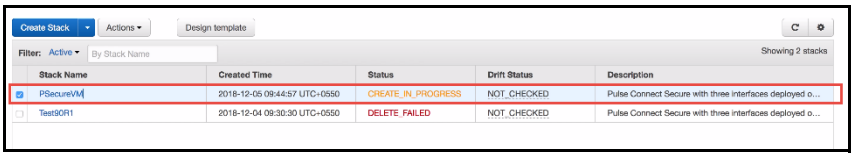


Review

1. In the Review page, verify the details and click **Create**.



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



To access Pulse Connect Secure Virtual Appliance, see "Accessing the Pulse Connect Secure Virtual Appliance" on page 40

Pulse Connect Secure on Amazon Web Services

Prerequisites and System Requirements on AWS

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

- An AWS account
- Access to the AWS portal (https://console.aws.amazon.com/)*
- Pulse Connect Secure Virtual Appliance Image (.ami file)
- AWS CloudFormation template / AWS Terraform template
- Pulse Connect Secure licenses **
- Site-to-Site VPN between AWS 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)

Note:

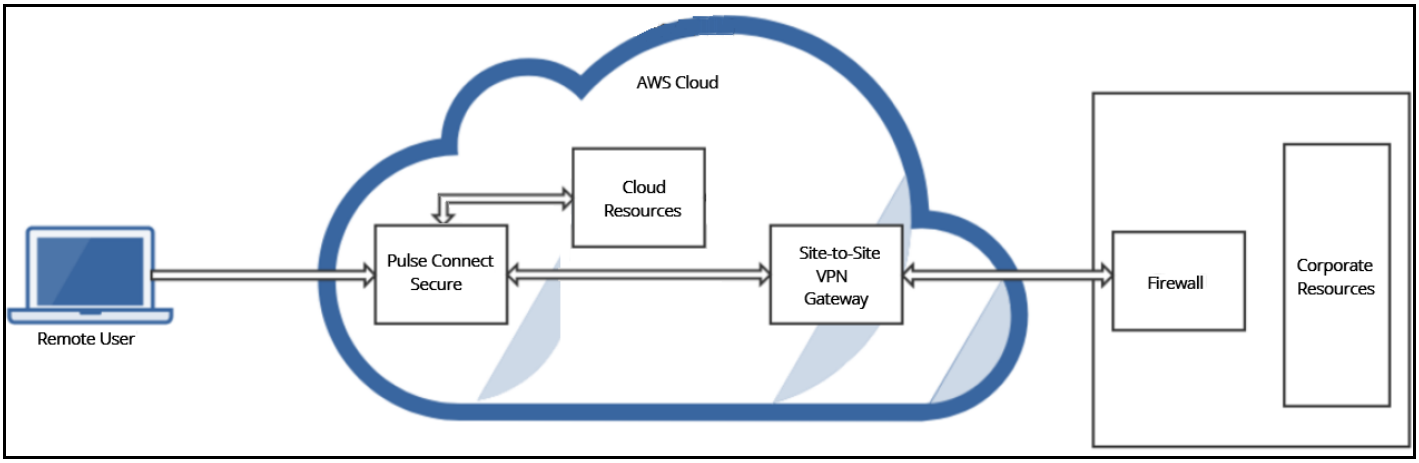
* Pulse Connect Secure Virtual Appliance can be deployed only through AWS CloudFormation style.

** From 9.0R3 release, Pulse Connect Secure Virtual Appliance, by default, has two evaluation licenses, and supports licensing with License server located at corporate network.

Deploying Pulse Connect Secure on Amazon Web Services

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 AWS and the corporate network.

Figure 6 Pulse Connect Secure on AWS



Supported Platform Systems

This section helps you in choosing the instance types that should be deployed with Pulse Connect Secure for AWS.

- PSA3000v is equivalent to t2 medium
- PSA5000v is equivalent to t2.xlarge
- PSA7000v is equivalent to t2.2xlarge

Model	vCPU	CPU Credits / hour	Memory (GiB)	Storage
t2.nano	1	3	0.5	EBS-Only
t2.micro	1	6	1	EBS-Only
t2.small	1	12	2	EBS-Only
t2.medium	2	24	4	EBS-Only
t2.large	2	36	8	EBS-Only
t2.xlarge	4	54	16	EBS-Only
t2.2xlarge	8	81	32	EBS-Only

Steps to Deploy Pulse Connect Secure on AWS

Registering the AMI

This section describes the steps to register the AMI. This is the one-time activity to be followed to deploy Pulse Connect Secure on AWS.

Prerequisites

- AWS command line should be configured on the host.
- the image should be available locally on the host.

To register AMI, do the following:

1. Download PCS Xen image which is in zip format from Pulse support site and unzip the file.
2. Install AWS CLI on the client machine. For the software and installation details, refer the link https://aws.amazon.com/cli/.
3. Copy PCS Xen image on the client machine.
4. Create Amazon S3 bucket and VM Import service role by following the procedures mentioned in https://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-image-import.html#vmimport-iam-permissions
5. Upload the PCS Xen image to AWS S3 bucket by typing the following command:

   ```
   aws s3 cp <image> s3://<bucket>/<folder>/<imagename>
   ```

   where, bucket and folders are created in the desired S3 location.
6. Create a snapshot by doing the following:
   1. Prepare a container json file by entering the details:

      ```
      $ cat container.json
         {
           "Description": "fill-description",
           "Format": "raw",
           "UserBucket": {
               "S3Bucket": "bucket-name-where-image-is-uploaded",
               "S3Key": " path of image: <folder>/<imagename>"
             }
           }
      ```

   2. After preparing container.json appropriately, run the following command:

```
aws ec2 import-snapshot --description "<description>" --disk-
container file:container.json --region <your-ec2-region>
```

This command will return a json file describing the status. Make a note of the "ImportTaskId" field from the json output.

3. Monitor the progress by running the following command:

```
aws ec2 describe-import-snapshot-tasks --region <your-ec2-region>
--import-task-ids <import-task-id>
```

Monitor the progress until the "status:Completed" message appears, and a snapshotId is added in the json output. Make note of the "SnapshotId".

7. Register an AMI from the snapshot by running the following command:

```
aws ec2 register-image --description "<description>" --
architecture x86_64 --name <image-name> --block-device-mappings
DeviceName="/dev/xvda",Ebs={SnapshotId=<snapshot-id>} --
virtualization-type hvm --root-device-name "/dev/xvda" --region
<your-ec2-region>
```

This completes AMI registration.

Deploying Pulse Connect Secure on AWS using AWS Portal

Once the access to the AMI file and CloudFormation template is obtained as mentioned in the above section, proceed with the Pulse Connect Secure deployment.

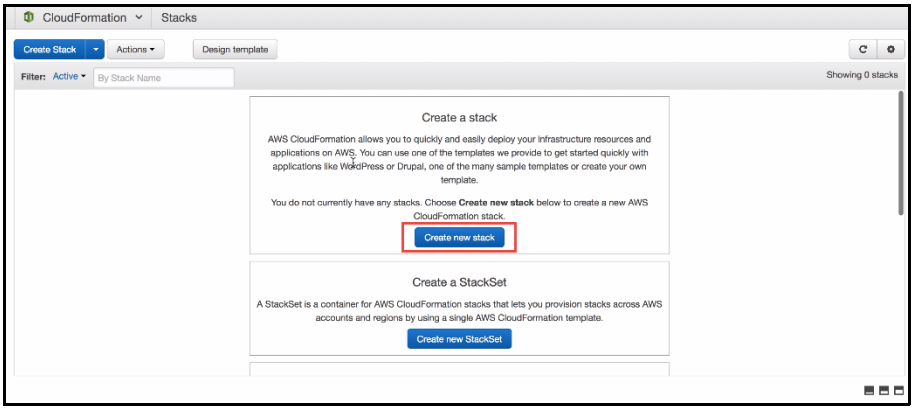
Deploying PCS on New Virtual Private Cloud

This section describes PCS deployment with "Deployment on VM with Three NIC Cards" on page 17 and "Deployment on VM with Two NIC Cards" on page 19.

Deployment on VM with Three NIC Cards

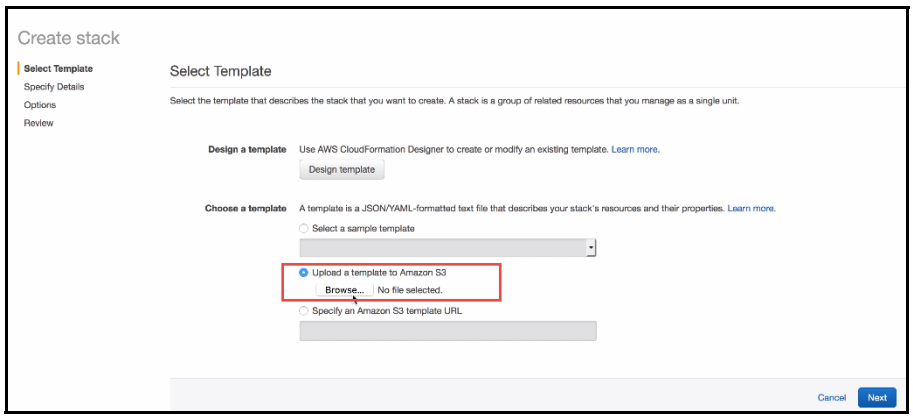
To deploy Pulse Connect Secure on AWS using the AWS portal, do the following:

1. Select **AWS Services > CloudFormation** and click **Create new stack**.
2. Create **New Stack**



3. Select **Upload a template to Amazon S3**. Click **Browse** and select "pulsesecure-PCS-3-nics-new-network.json" template file for the new VPC. Then click **Next**.

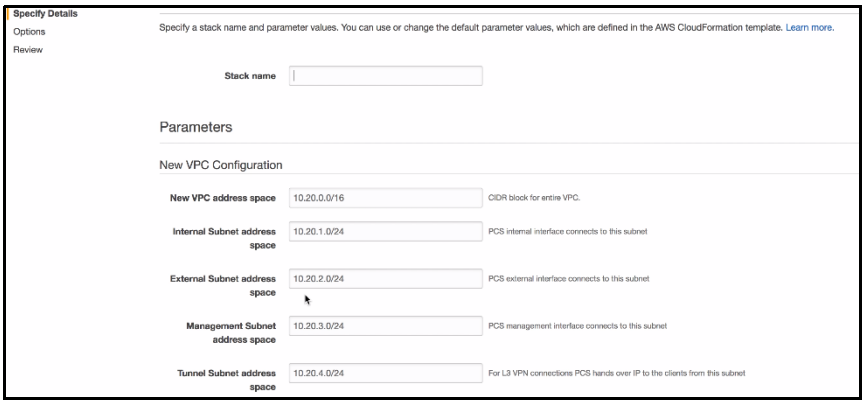
Figure 7 Upload Template



4. In the Specify Details page, fill or modify the following parameters.

Note: Before proceeding with deployment, ensure that the attribute "**accept-license-agreement" i**n PCS ConfigData is set to "**y**".

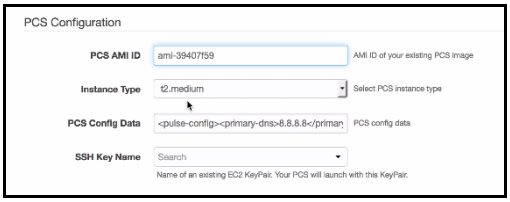
Figure 8 Specify Details for New Virtual Private Cloud.



- **Stack name:** Specify the stack name in which Pulse Connect Secure needs to be deployed
- **New VPC address space:** Virtual private cloud address space
- **Internal Subnet address space:** Subnet from which Pulse Connect Secure internal interface needs to lease IP
- **External Subnet address space:** Subnet from which Pulse Connect Secure external interface needs to lease IP
- **Management Subnet address space:** Subnet from which Pulse Connect Secure management interface needs to lease IP
- **Tunnel Subnet address space**: Subnet which will be configured as tunnel IP pool in the Pulse Connect Secure VPN Profile
- **PCS AMI ID**: ID of the uploaded AMI file
- **Instance Type:** Size of the instance – t2.medium or t2.large

- **PCS Config Data:** Provisioning parameters in an XML format. For details, see Pulse Connect Secure Provisioning Parameters.
- **SSH Key Name:** 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 http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html.
- Review the specified details and click **Create**. Observe the deployed PCS in a few minutes.

Figure 9 New VPC

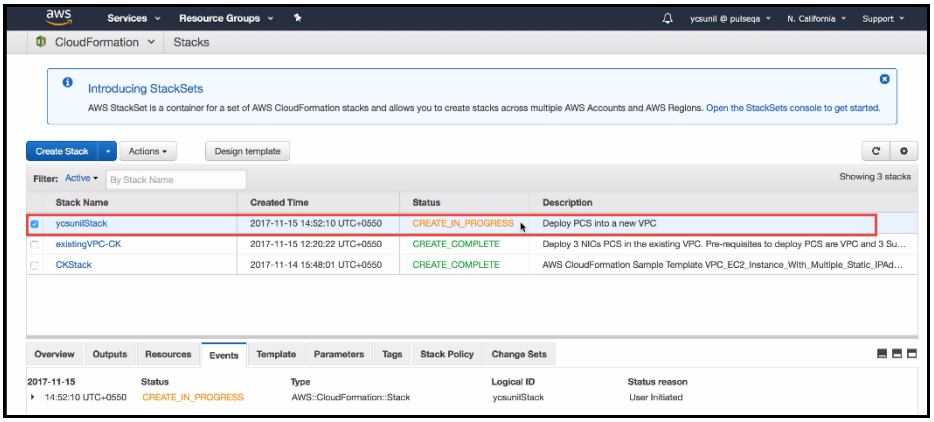


Deployment on VM with Two NIC Cards

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

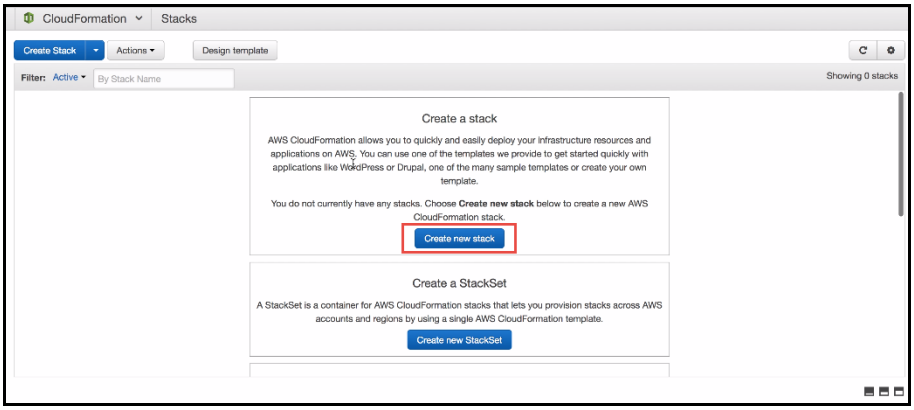
1. Select **AWS Services > CloudFormation** and click **Create new stack**.

Figure 10 Create New Stack.



2. Select **Upload a template to Amazon S3**. Click **Browse** and select "pulsesecure-pcs-2-nics-new-network.json" template file for the new VPC. Then click **Next**.

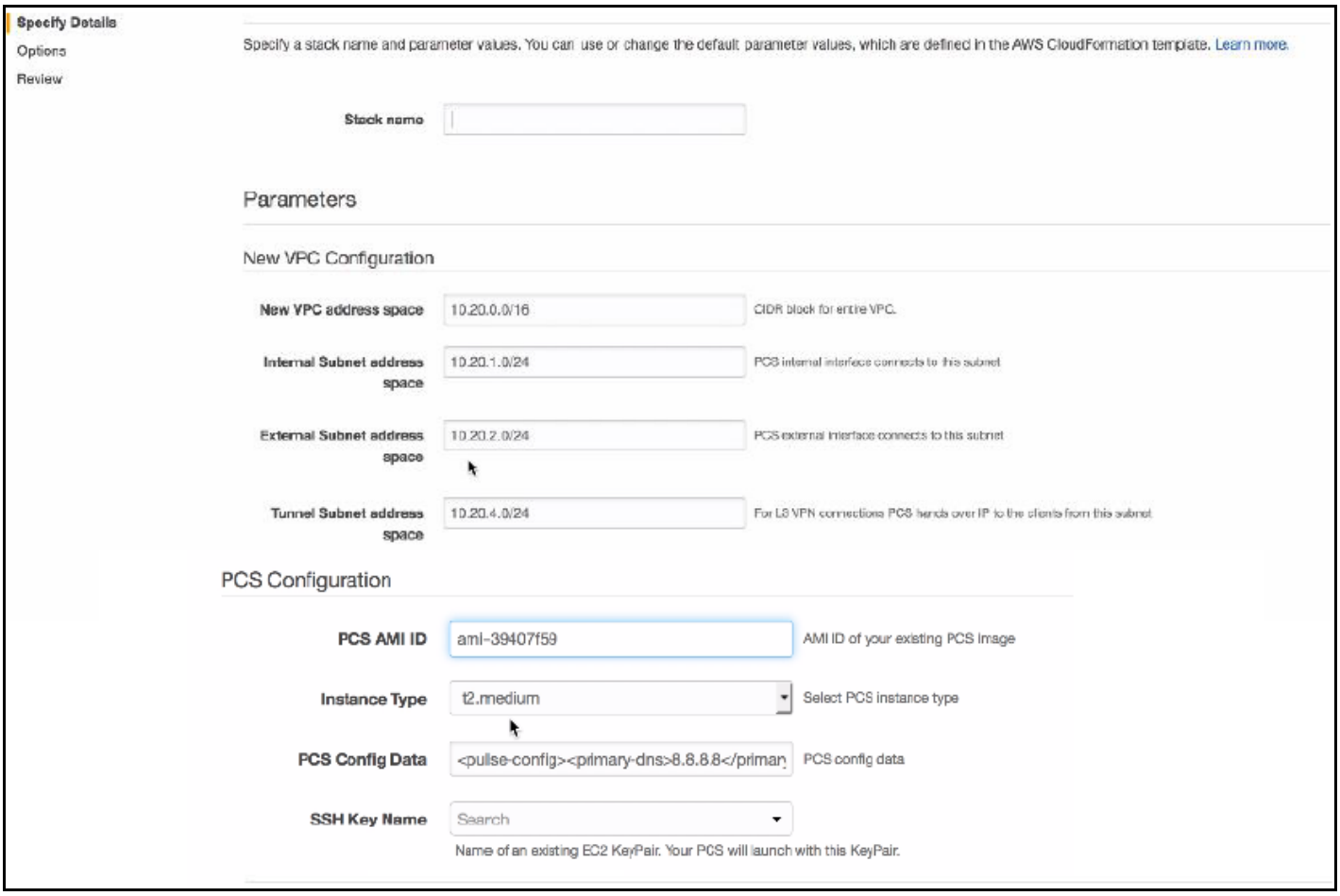
Figure 11 Upload Template



3. In the Specify Details page, fill or modify the following parameters.

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

Figure 12 Specify Details for New Virtual Private Cloud

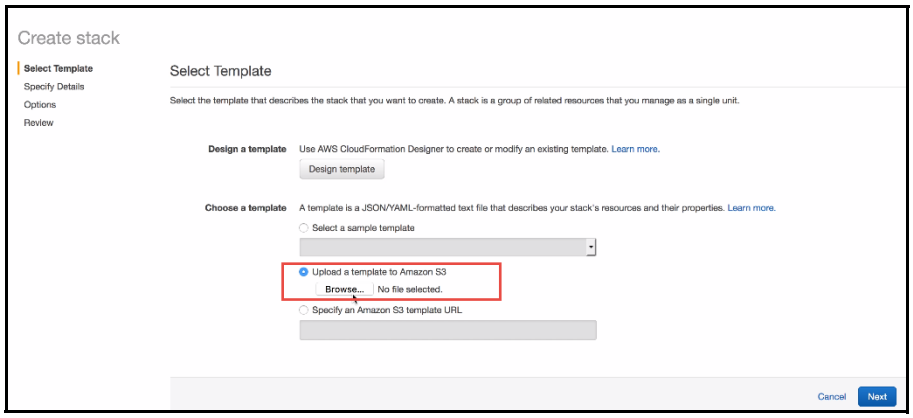


- **Stack name**: Specify the stack name in which Pulse Connect Secure needs to be deployed
- **New VPC address space:** Virtual private cloud address space
- **Internal Subnet address space:** Subnet from which Pulse Connect Secure internal interface needs to lease IP
- **External Subnet address space:** Subnet from which Pulse Connect Secure external interface needs to lease IP
- **Tunnel Subnet address space:** Subnet which will be configured as tunnel IP pool in the Pulse Connect Secure VPN Profile
- **PCS AMI ID:** ID of the uploaded AMI file
- **Instance Type:** Size of the instance – t2.medium or t2.large

- **PCS Config Data:** Provisioning parameters in an XML format. For details, see "Pulse Connect Secure Provisioning Parameters" on page 34.
- **SSH Key Name:** 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 http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html.

4. Review the specified details and click **Create**. Observe the deployed PCS in a few minutes.

Figure 13 New VPC



Deploying PCS on an Existing Virtual Private Cloud

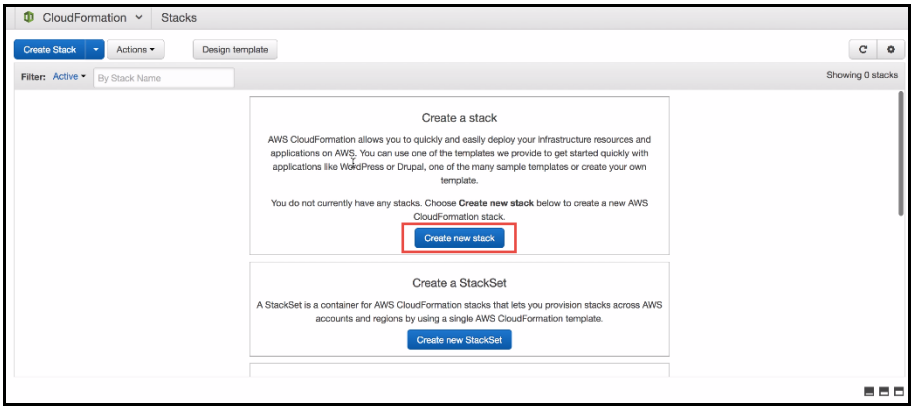
This section describes PCS deployment with "Deployment on VM with Three NIC Cards" on page 21 and "Deployment on VM with Two NIC Cards" on page 23.

Deployment on VM with Three NIC Cards

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

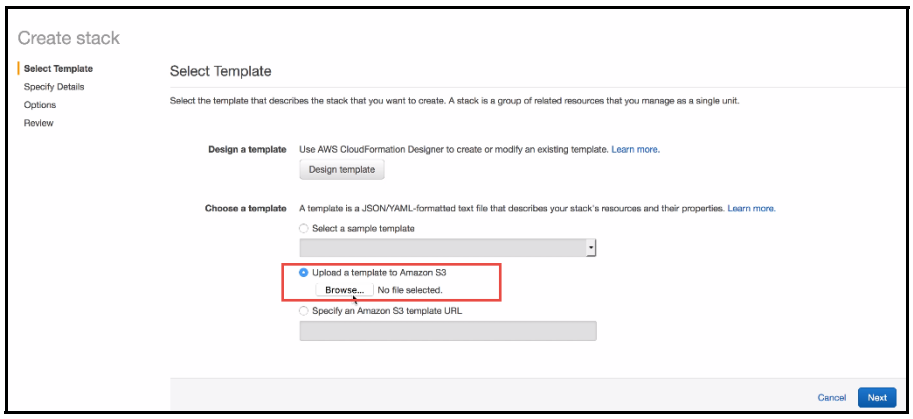
1. Select **AWS Services > CloudFormation** and click **Create new stack**.

Figure 14 Create New Stack



2. Select **Upload a template to Amazon S3**. Click **Browse** and select "pulsesecure-PCS-3-nics-existing-vpc.json" template file for existing VPC. Then click **Next**.

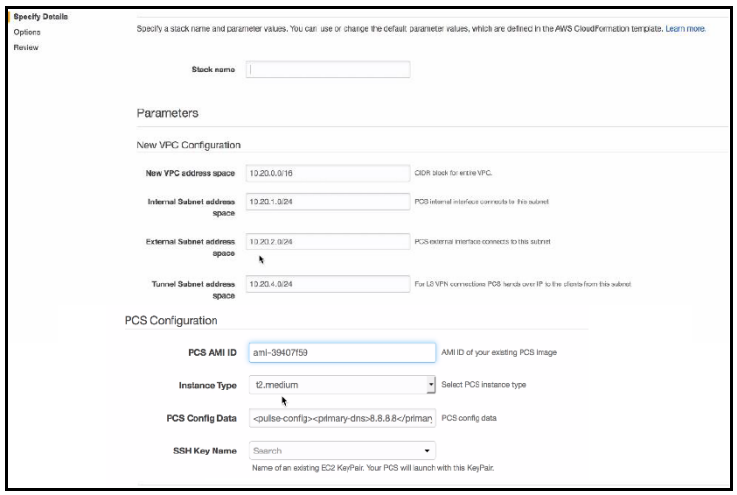
Figure 15 Upload Template

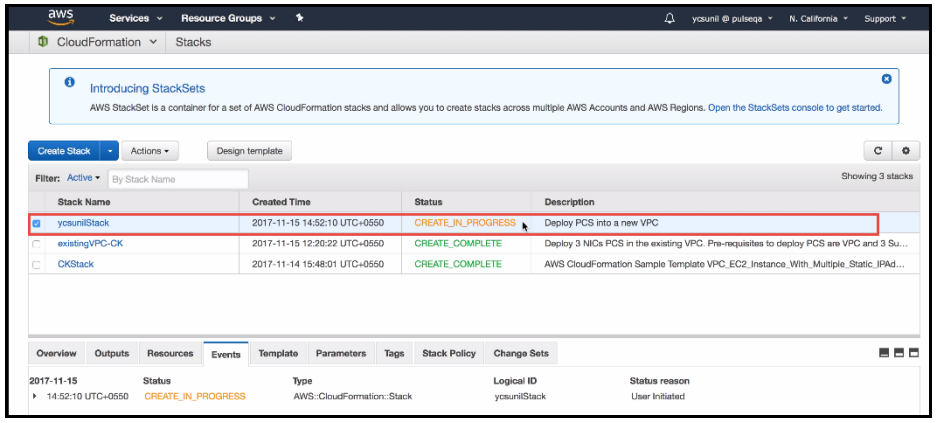


3. In the Specify Details page, fill or modify the following parameters.

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

Figure 16 Specify Details for Existing Virtual Private Cloud





- **Stack name:** Specify the stack name in which Pulse Connect Secure needs to be deployed

- **Existing VPC ID:** Virtual private cloud ID
- **Internal Subnet ID:** Subnet from which Pulse Connect Secure internal interface needs to lease IP
- **External Subnet ID:** Subnet from which Pulse Connect Secure external interface needs to lease IP
- **Management Subnet ID:** Subnet from which Pulse Connect Secure management interface needs to lease IP
- **Tunnel Subnet ID:** Subnet which will be configured as tunnel IP pool in the Pulse Connect Secure VPN Profile
- **PCS AMI ID:** ID of the uploaded AMI file
- **Instance Type:** Size of the instance – t2.medium or t2.large
- **PCS Config Data**: Provisioning parameters in an XML format. For details, see "Pulse Connect Secure Provisioning Parameters" on page 34.
- **SSH Key Name:** 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 http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html

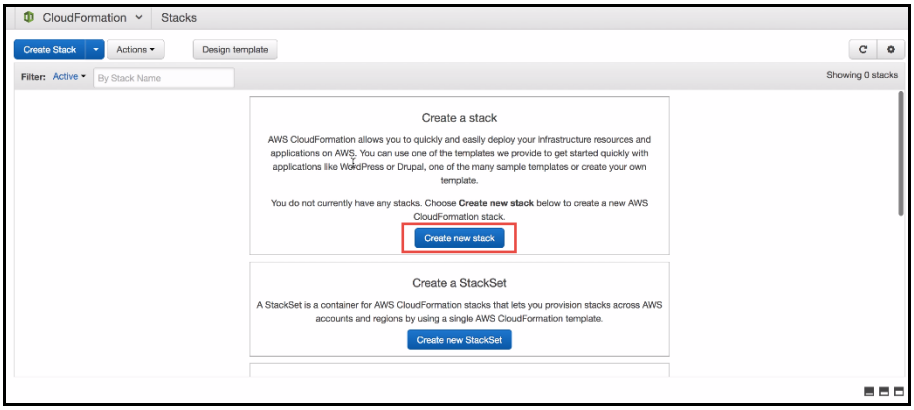
4. Review the specified details and click **Create**. Observe the deployed PCS in a few minutes.

Deployment on VM with Two NIC Cards

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

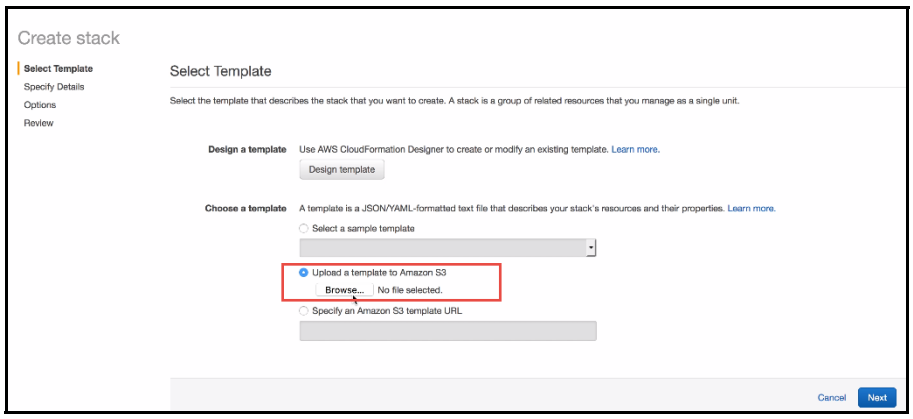
1. Select **AWS Services > CloudFormation** and click **Create new stack**.

   Figure 17 Create New Stack



2. Select **Upload a template to Amazon S3**. Click **Browse** and select "pulsesecure-pcs-2-nics-existing-vpc.json" template file for existing VPC. Then click **Next**.

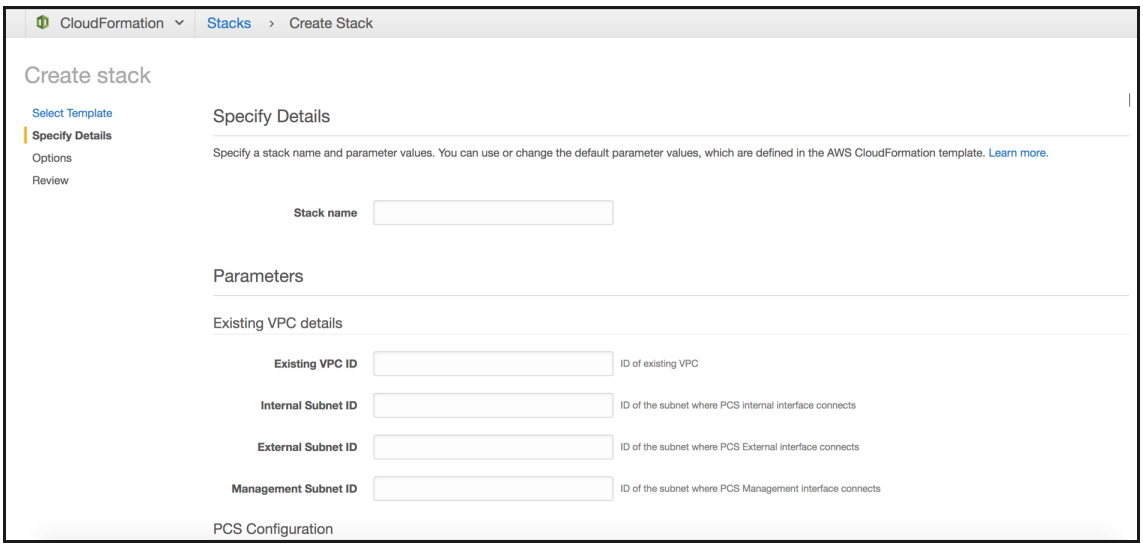
Figure 18 Upload Template



3. In the Specify Details page, fill or modify the following parameters.

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

   Figure 19 Specify Details for Existing Virtual Private Cloud



- **Stack name:** Specify the stack name in which Pulse Connect Secure needs to be deployed
- **Existing VPC ID:** Virtual private cloud ID
- **Internal Subnet ID:** Subnet from which Pulse Connect Secure internal interface needs to lease IP
- **External Subnet ID:** Subnet from which Pulse Connect Secure external interface needs to lease IP
- **Tunnel Subnet ID**: Subnet which will be configured as tunnel IP pool in the Pulse Connect Secure VPN Profile
- **PCS AMI ID**: ID of the uploaded AMI file
- **Instance Type:** Size of the instance – t2.medium or t2.large
- **PCS Config Data:** Provisioning parameters in an XML format. For details, see Pulse Connect Secure Provisioning Parameters.

- **SSH Key Name:** 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 *http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html*

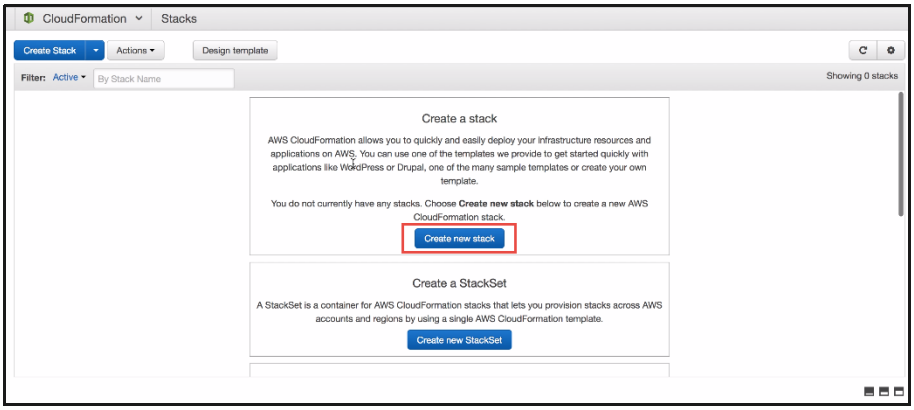
4. Review the specified details and click **Create**. Observe the deployed PCS in a few minutes.

Deploying PCS as a License Server

To deploy Pulse Connect Secure on AWS as a license server, do the following:

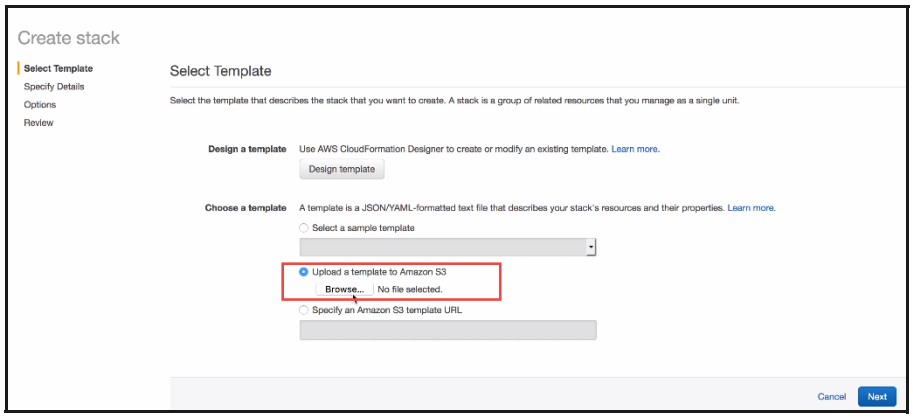
1. Select **AWS Services > CloudFormation** and click **Create new stack**.

   Figure 20 Create New Stack



2. Select **Upload a template to Amazon S3**. Click **Browse** and select "pulsesecure-pcs-3-nics-existing-vpc.json" template file for existing VPC. Then click **Next**.

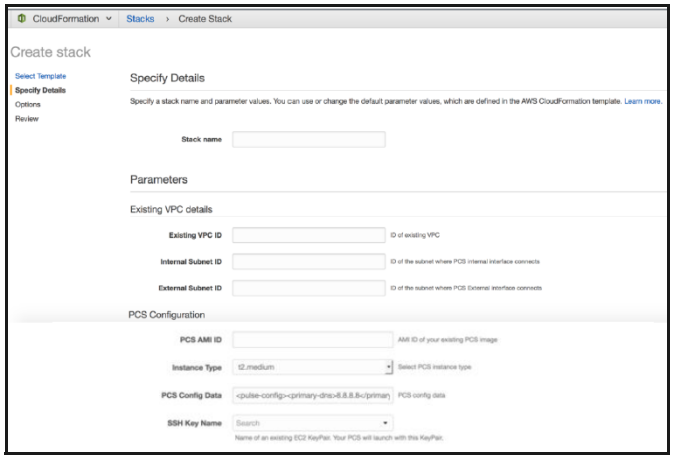
   Figure 21 Upload Template

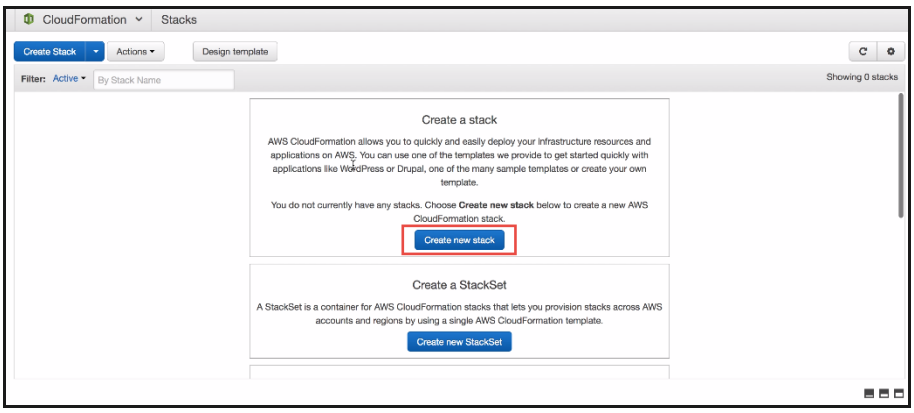


3. In the Specify Details page, edit the **PCS Config Data** text box to enable PCS as license server by setting the **enable-license-server** attribute to **y** as follows.

   <enable-license-server>y</enable-license-server>

Figure 22 Specify Details for Existing Virtual Private Cloud

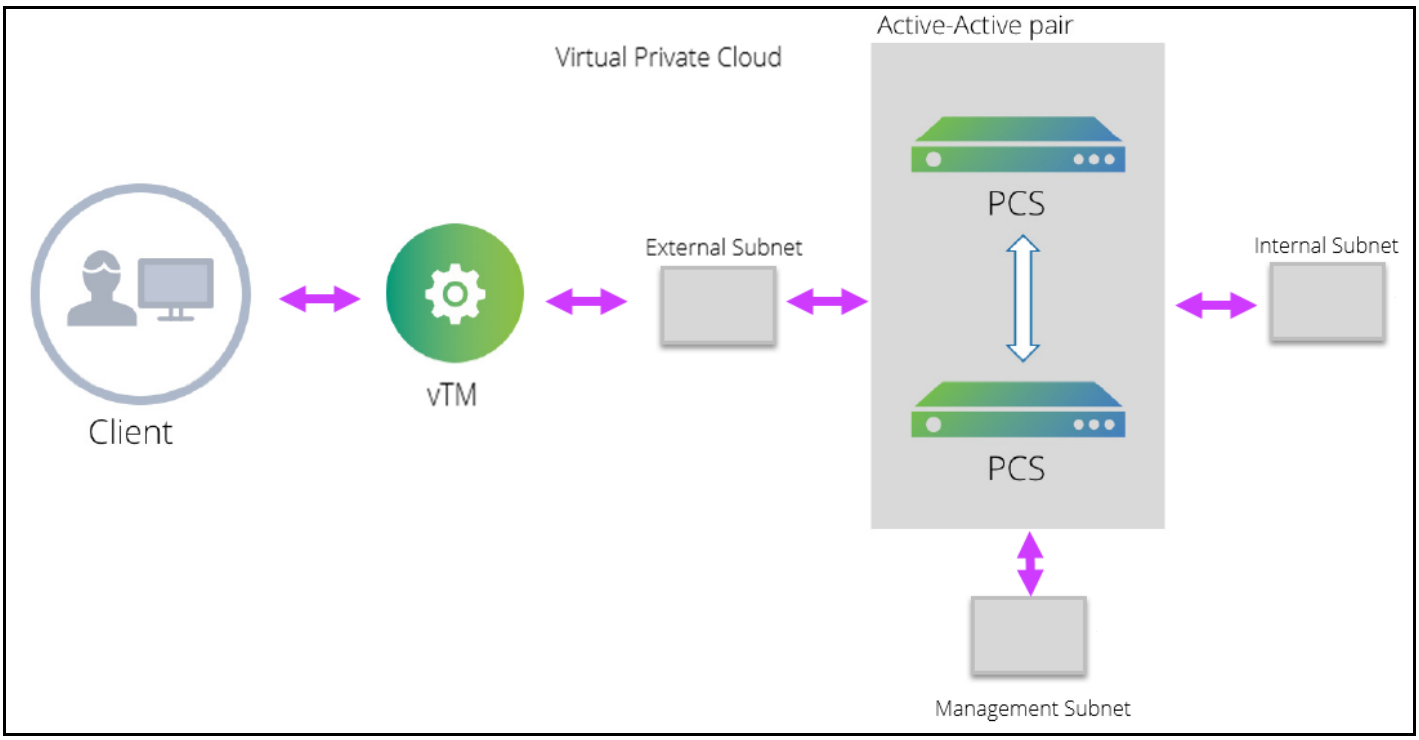




Deploying PCS Active-Active Cluster using Virtual Traffic Manager in AWS

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

Figure 23 Deploying PCS A-A Cluster Topology Diagram



The deployment process involves the following steps:

- "Deploying Two PCS EC2 instances Using CloudFormation Template" on page 27
- "Forming the Active-Active Cluster" on page 27
- "Deploying Virtual Traffic Manager EC2 Instance in the External Subnet of PCS in AWS" on page 28
- "Setting Up and Configuring vTM for External Users" on page 30

Deploying Two PCS EC2 instances Using CloudFormation Template

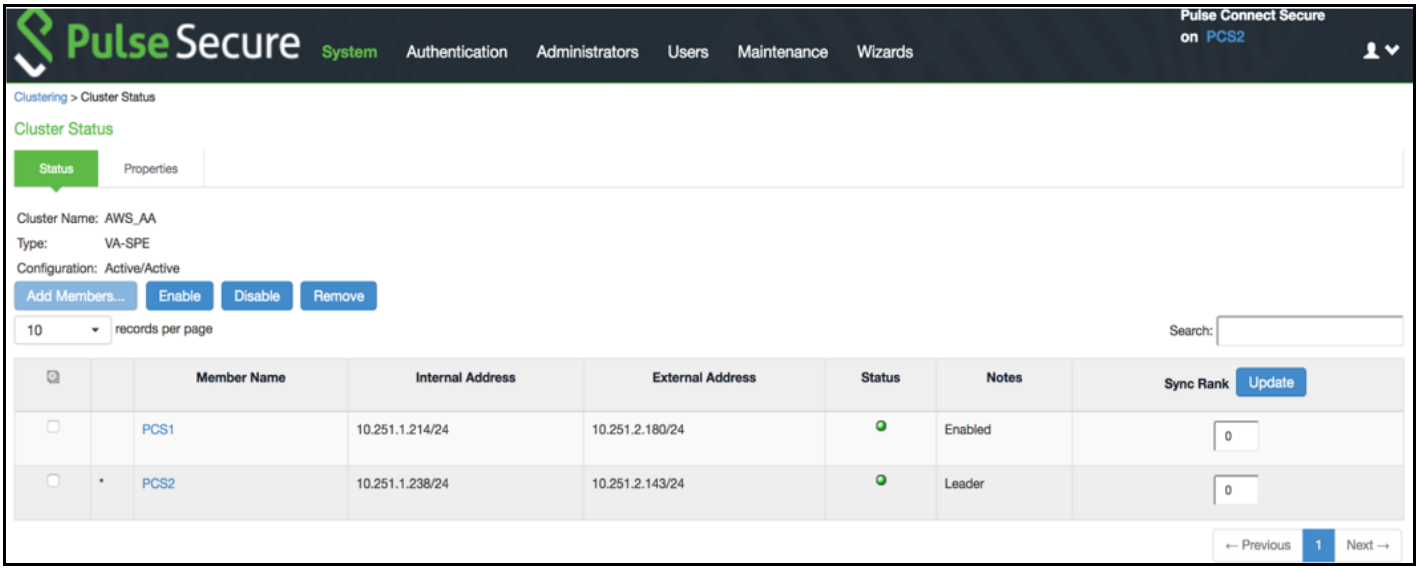
PCS can be deployed in AWS using CloudFormation template in a 3-armed model. Based on the need, deploy two PCS instances using one of the following templates:

- pulsesecure-pcs-3-nics-new-network.json
- pulsesecure-pcs-3-nics-existing-vpc.json

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.

Figure 24 PCS A-A Cluster Status



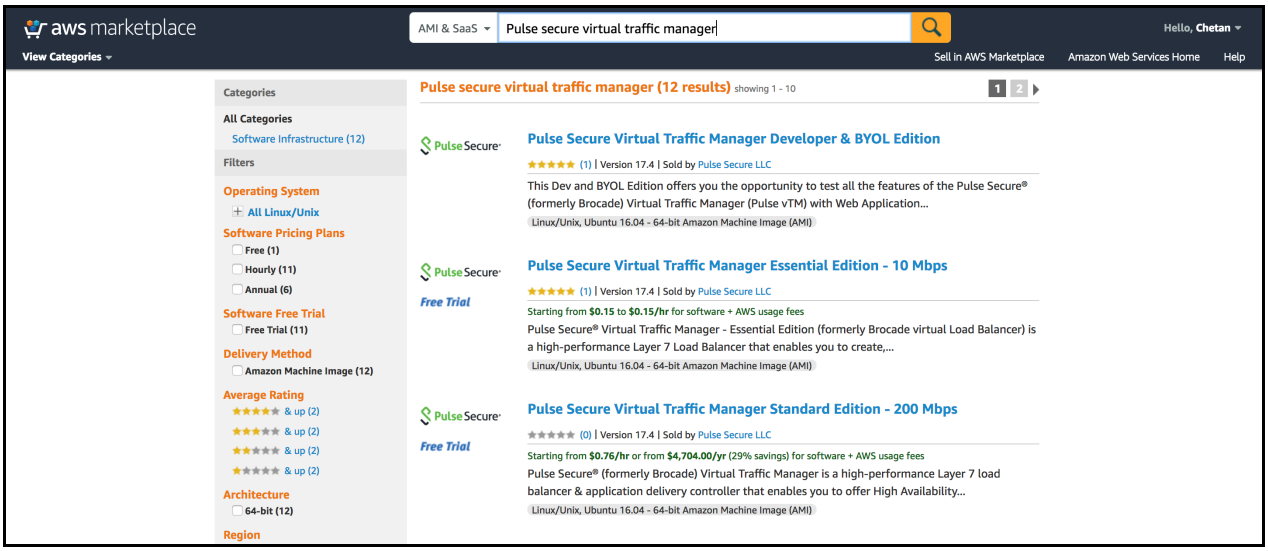
Deploying Virtual Traffic Manager EC2 Instance in the External Subnet of PCS in AWS

Virtual Traffic Manager can be deployed through either AWS Marketplace or AWS CLI.

To deploy through Marketplace, follow the below steps:

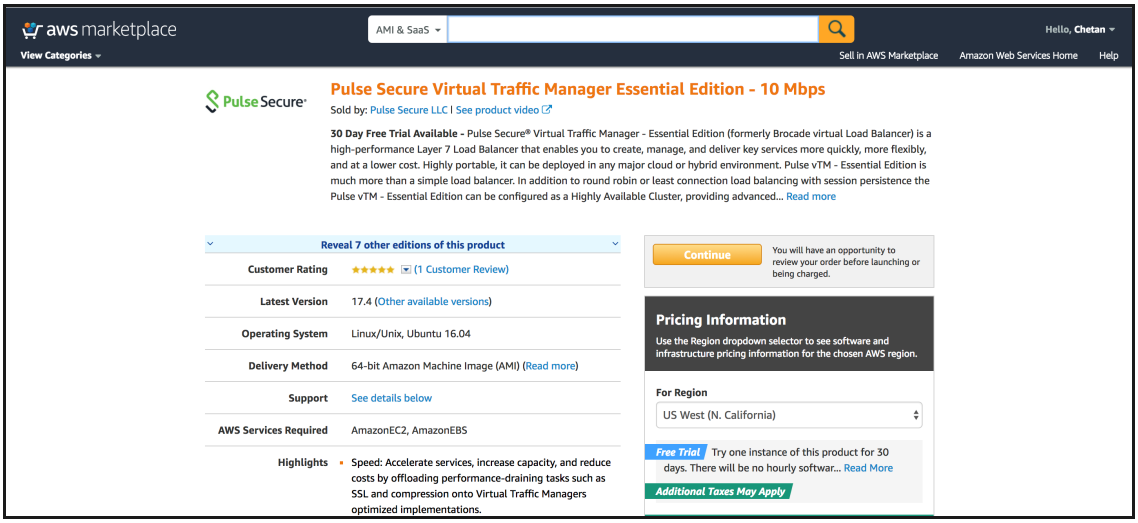
1. Search and select **Pulse Secure vTM** in AWS Marketplace.

   Figure 25 AWS Marketplace > Pulse Secure vTM



2. Select the required vTM variant and AWS region, and click **Continue**.

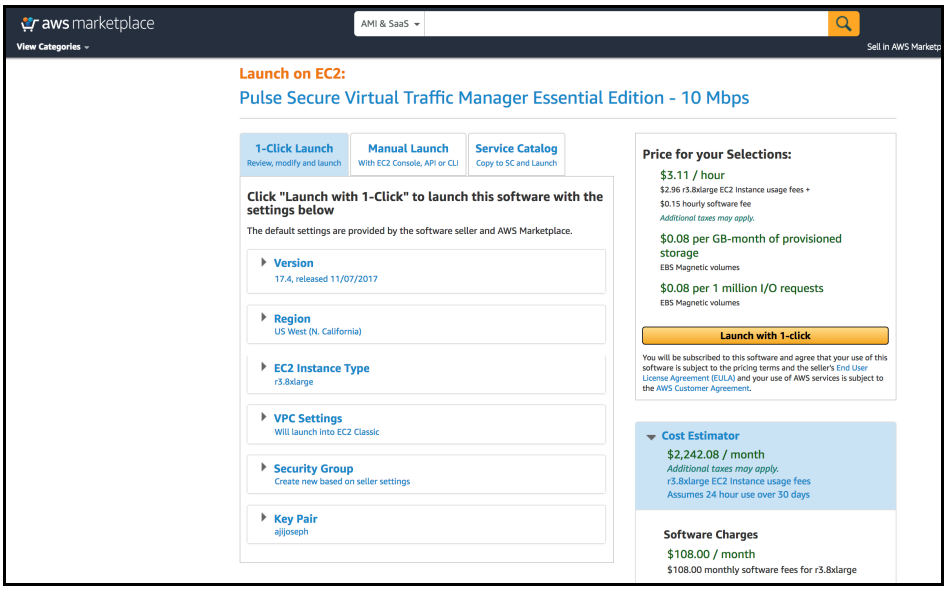
Figure 26 vTM Editions Available in AWS Marketplace



3. Under the **1-Click Launch** tab, update the following required details:

   - EC2 instance type
   - VPC setting
   - Security group
   - In the **VPC Settings** tab, select the VPC and subnet matching PCS's VPC and external subnet.

Figure 27 Click Launch Tab



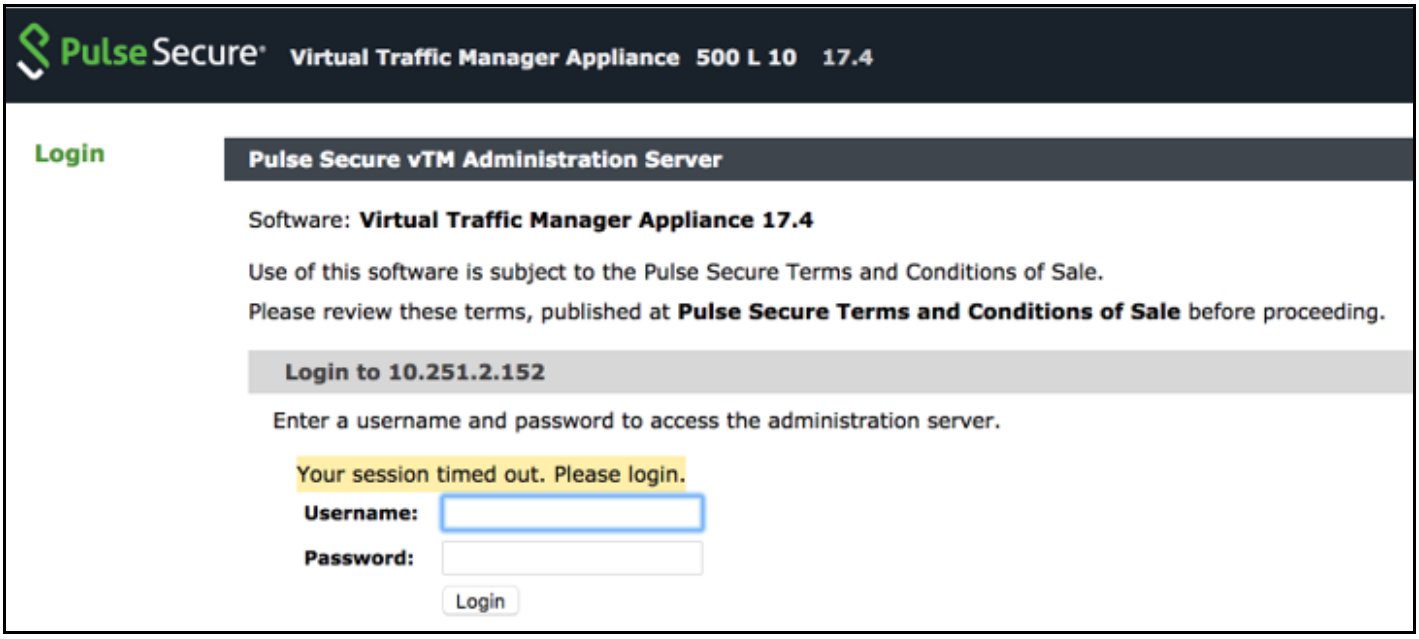
To deploy vTM though AWS CLI, follow the steps in the section "Creating a Traffic Manager Instance on Amazon EC2" in Pulse Secure Virtual Traffic Manager: Cloud Services Installation and Getting Started Guide. Make sure that vTM is deployed on the external network of PCS.

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 28 Pulse Secure vTM Login Page



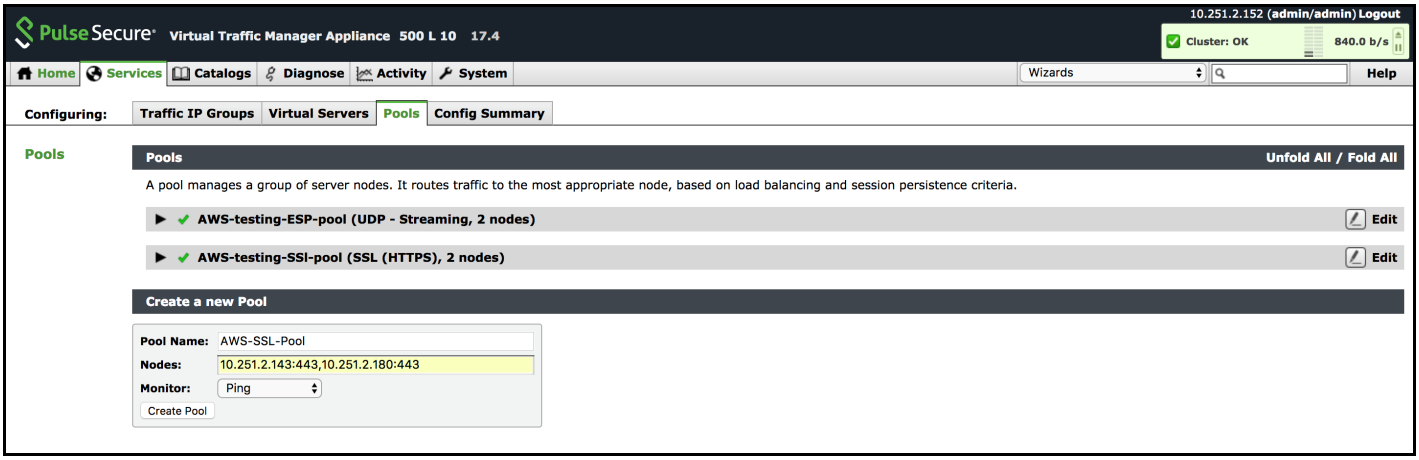
Next step is to set up the vTM for the external users using traffic pools and load balancing virtual servers. Traffic pool is the group that will bind to virtual server for load balancing. In an Active-Active Cluster scenario, traffic pool comprises cluster nodes. We need to create two separate traffic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

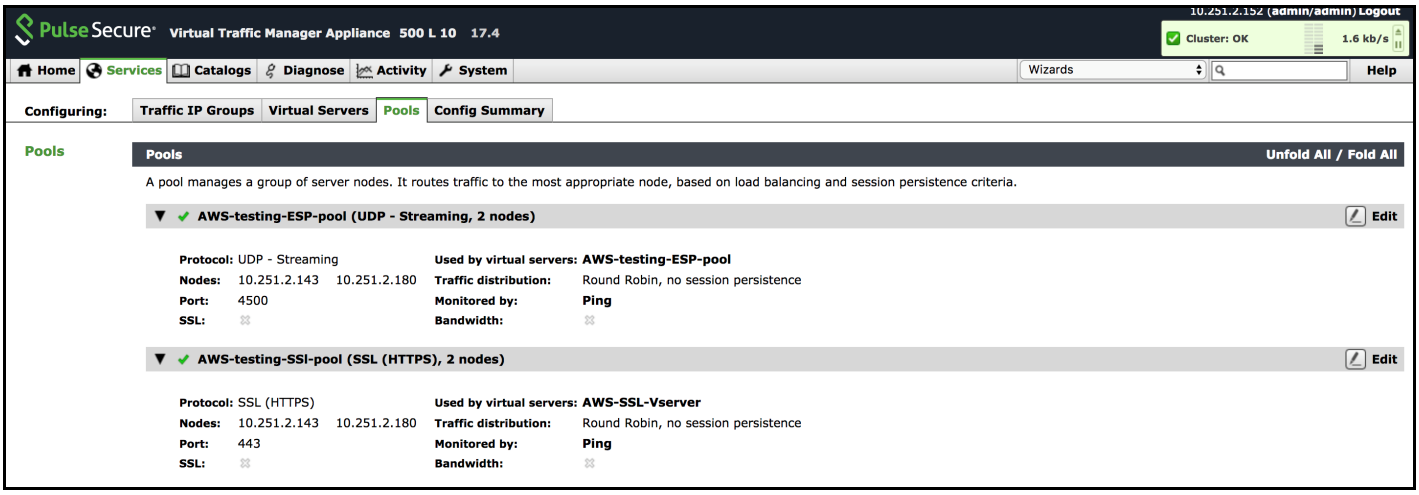
Complete these steps for SSL and UDP. For details, refer to the section “Creating PCS Pools” in Load Balancing PCS with vTM Deployment Guide.

Figure 29 Create Traffic Pool



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

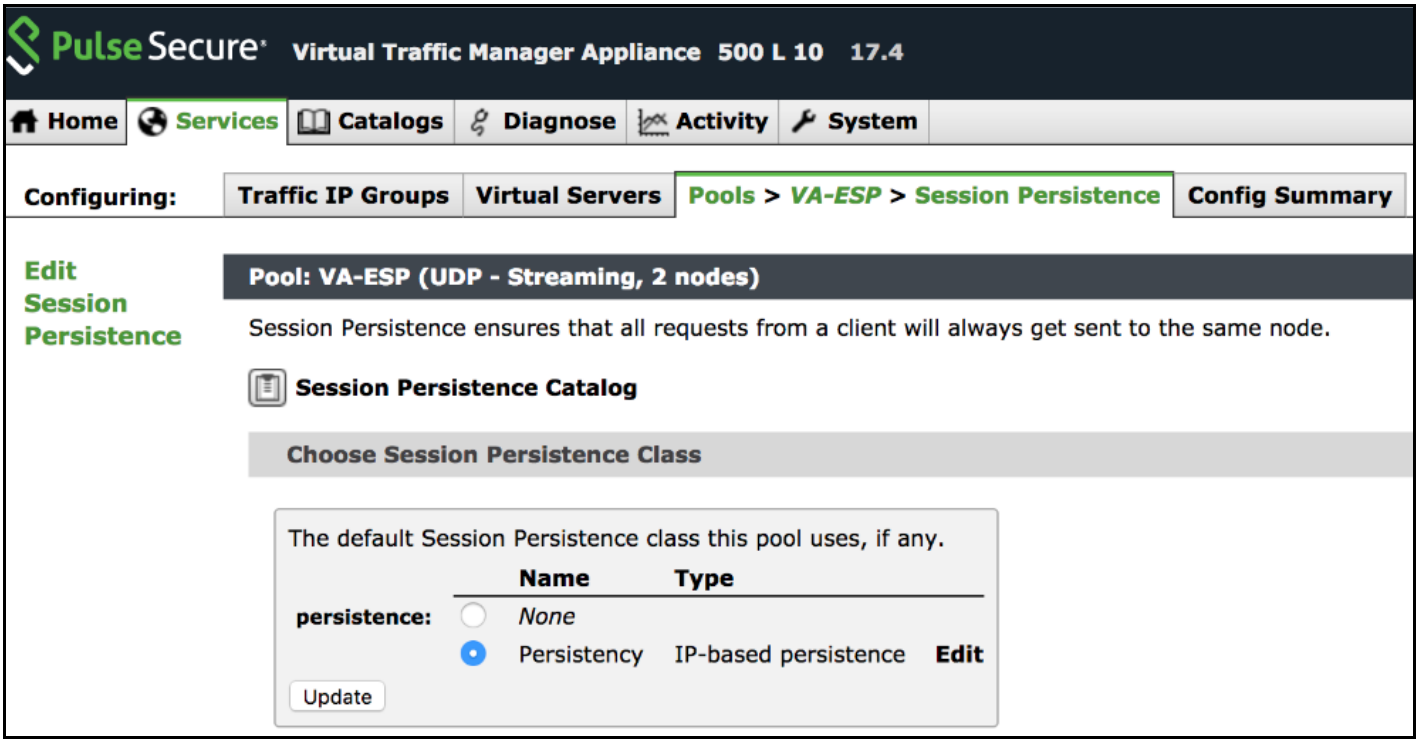
Figure 30 SSL and UDP Pools



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 31 Session Persistency Class



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 32 Create Virtual Server

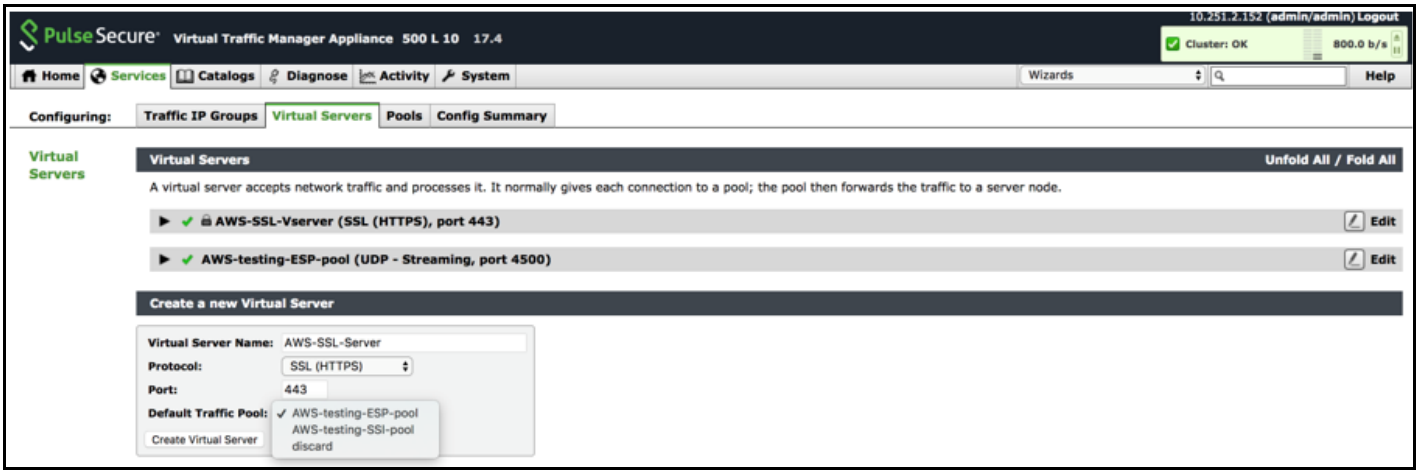
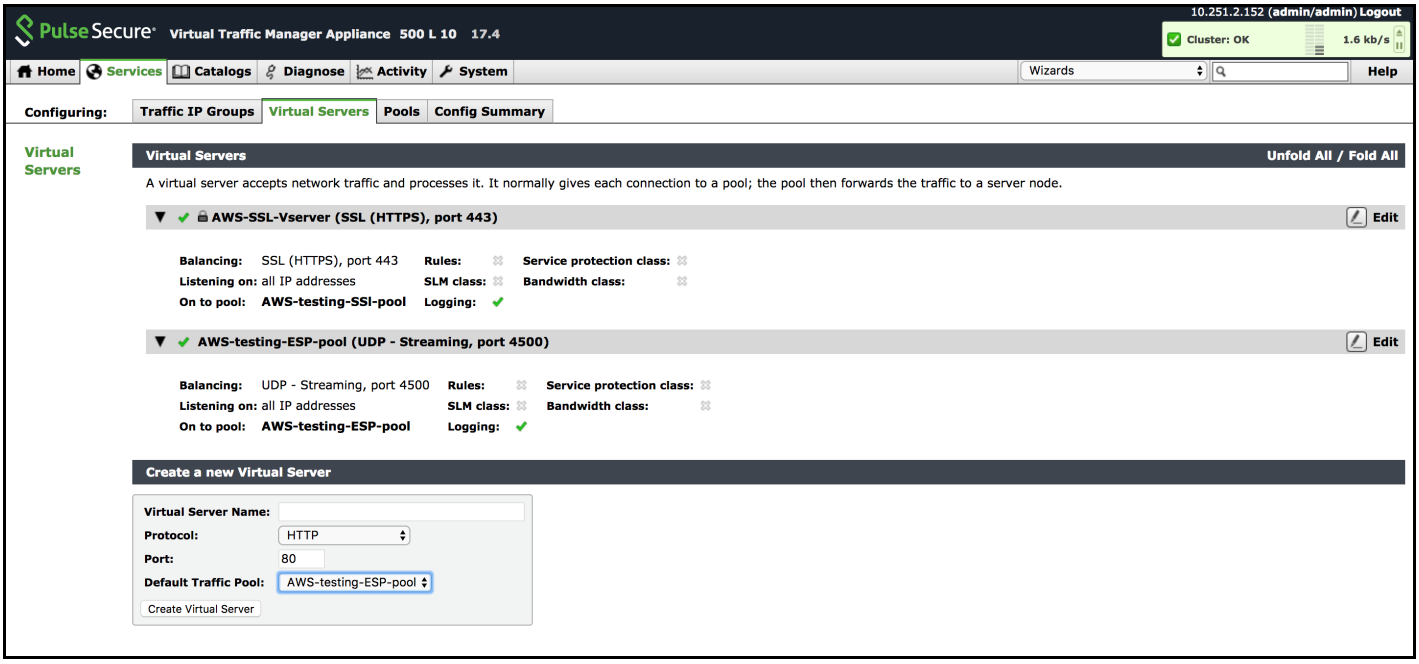
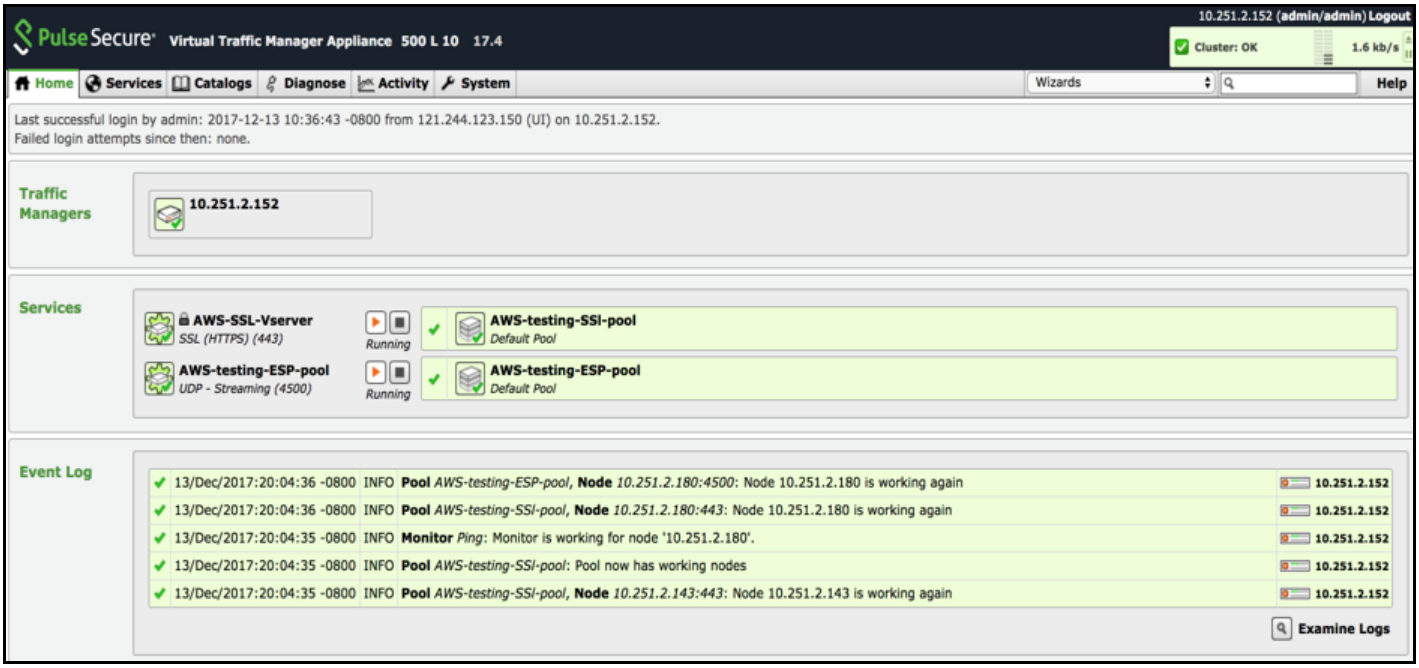


Figure 33 Virtual Servers to Handle SSL and UDP Traffic



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

Figure 34 Pulse Secure vTM Home Page Showing Services and Event Logs



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><value></primary-dns>
    <secondary-dns><value></secondary-dns>
    <wins-server><value></wins-server>
    <dns-domain><value></dns-domain>
    <admin-username><value></admin-username>
    <admin-password><value></admin-password>
    <cert-common-name><value></cert-common-name>
    <cert-random-text><value></cert-random-text>
    <cert-organisation><value></cert-organisation>
    <config-download-url><value></config-download-url>
    <config-data><value></config-data>
    <auth-code-license><value></auth-code-license>
    <enable-license-server><value></enable-license-server>
    <accept-license-agreement><value></accept-license-agreement >
    <enable-rest><value></enable-rest>
</pulse-config>
```

The below table depicts the details of the 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
8	cert-random-text	string	Random text for the self-certificate generation
9	cert-organization	string	Organization name for the self-signed certificate generation

#	Parameter Name	Type	Description
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 AWS cloud or at corporate network which is accessible for Pulse Connect Secure through site to site VPN between AWS 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. 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.

Note: The XML parsing fails if the following characters are used in the strings:

- "
- '
- <
- >
- &

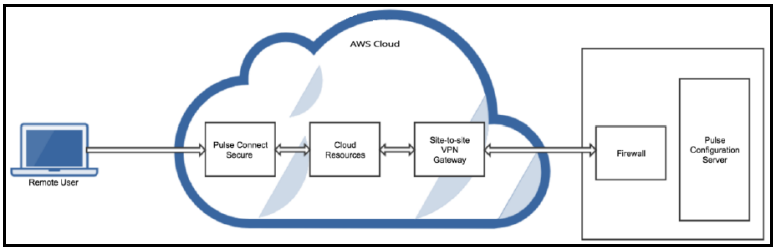
Note: From 9.1R3 release, Pulse Connect Secure supports zero touch provisioning. This feature can detect and assign DHCP networking settings automatically at the Pulse Connect Secure boot up. The Pulse Connect Secure parameters should be set to null in order to fetch the networking configuration automatically from the DHCP server.

Provisioning Pulse Connect Secure with Predefined Configuration

The Pulse Connect Secure Virtual Appliance can be provisioned on AWS with a predefined 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" on page 34 for details about the Pulse Connect Secure specific provisioning parameters.
  Pulse Connect Secure configuration can be kept on AWS 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 AWS 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 CloudFormation template.

Figure 35 Pulse Configuration Server in Corporate Network

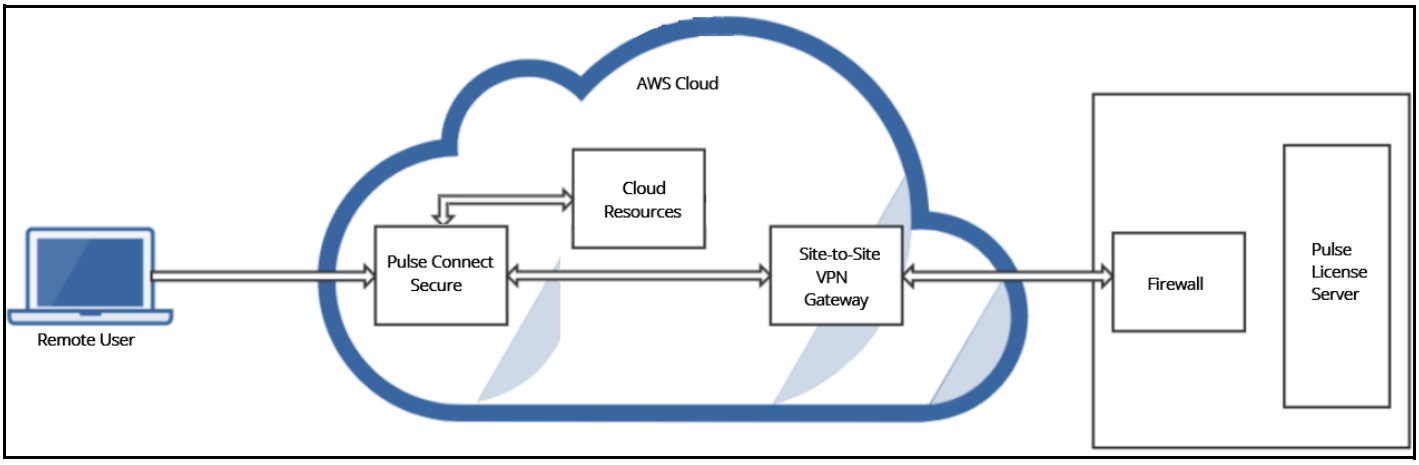


Configuring Licenses on the Pulse Connect Secure Appliance

In this release, evaluation licenses are provided. To add more licenses, the Pulse Connect Secure administrator needs to leverage the Pulse License server.

Pulse License Server in Corporate Network

Figure 36 Pulse License Server in a Corporate Network



Pulse License Server in Cloud Network

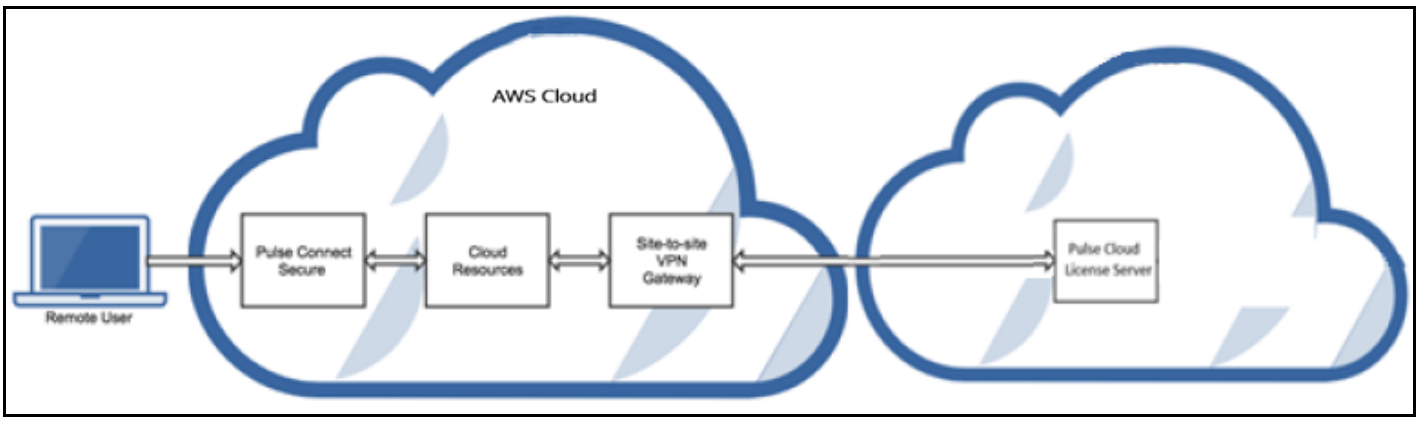
In 8.3R3, the Pulse Connect Secure virtual machines (VM) are enabled to provision licenses through the Pulse Cloud Licensing Service (PCLS). For this, administrator needs to obtain an Authentication code from Pulse Secure Support and apply it in Download Licenses page of PCS admin console. The PCS also periodically sends heartbeat messages to PCLS for auditing purposes.

```
"<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></config-download-url><config-data></config-data><auth-code-license></auth-code-license><enable-license-server>n</enable-license-server><accept-license-agreement>n</accept-license-agreement></pulse-config>"
```

The Authentication code can also be specified in the CloudFormation template. When PCS comes up, it automatically fetches the Authentication code.

- "Adding Authentication Code in PCS Admin Console" on page 38
- "Including Authentication Code in CloudFormation Template" on page 38

Figure 37 Pulse License Server in Cloud Network

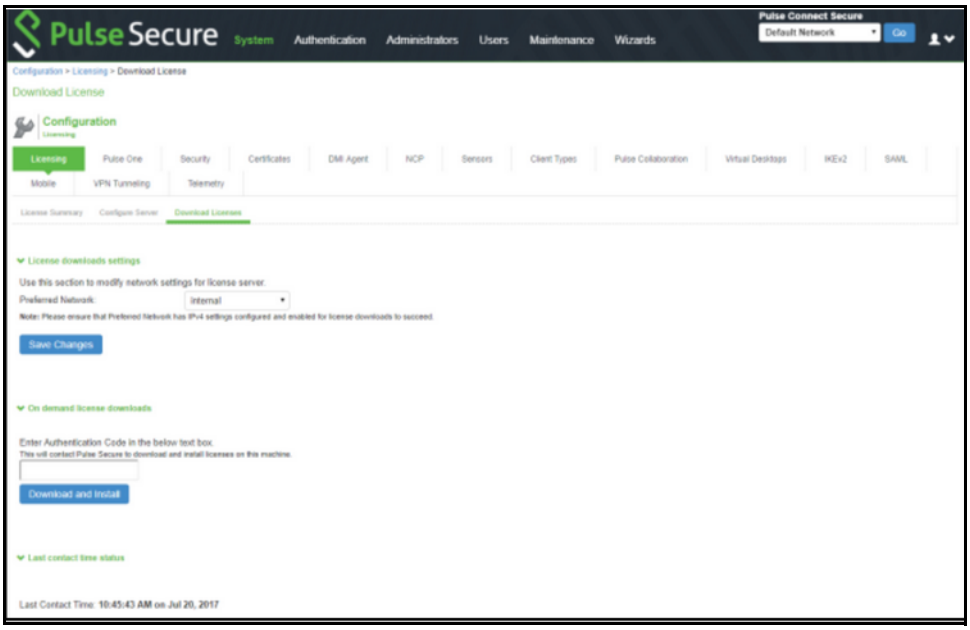


Adding Authentication Code in PCS Admin Console

To add Authentication code:

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

Figure 38 Enter Authentication Code



Including Authentication Code in CloudFormation Template

To include Authentication code in the CloudFormation template:

a. In the CloudFormation template, go to the PCSConfig section.
b. For the element <auth-code-license>, enter the Authentication code as the content.
c. Save the template.

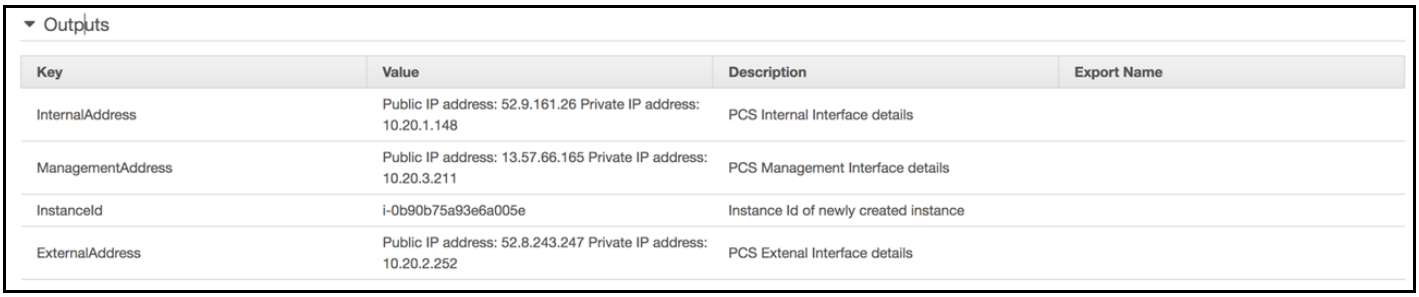
For details about the license configuration, refer to License Configuration Guide.

Accessing the Pulse Connect Secure Virtual Appliance

Accessing the Pulse Connect Secure Virtual Appliance as an Administrator

In the AWS portal, navigate to CloudFormation section. Select the stack where PCS is deployed and then click on the 'Outputs' tab. Note down the PCS management, internal and external address from the table as shown in Figure 39

Figure 39 Accessing PCS Virtual Appliance



▾ Outputs

Key	Value	Description	Export Name
InternalAddress	Public IP address: 52.9.161.26 Private IP address: 10.20.1.148	PCS Internal Interface details	
ManagementAddress	Public IP address: 13.57.66.165 Private IP address: 10.20.3.211	PCS Management Interface details	
InstanceId	i-0b90b75a93e6a005e	Instance Id of newly created instance	
ExternalAddress	Public IP address: 52.8.243.247 Private IP address: 10.20.2.252	PCS Extenal Interface details	

Use the credentials provided in the provisioning parameters to log in as the administrator in the PCS Admin interface with URL https://<PCS-IP>/admin. The default PCS Admin UI user configured in the CloudFormation config file is: user 'admin' and password 'password1234'.

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, AWS network administrator needs to ensure that all other resources have Pulse Connect Secure Internal interface as its default gateway.

Accessing the Pulse Connect Secure Virtual Appliance as an End User

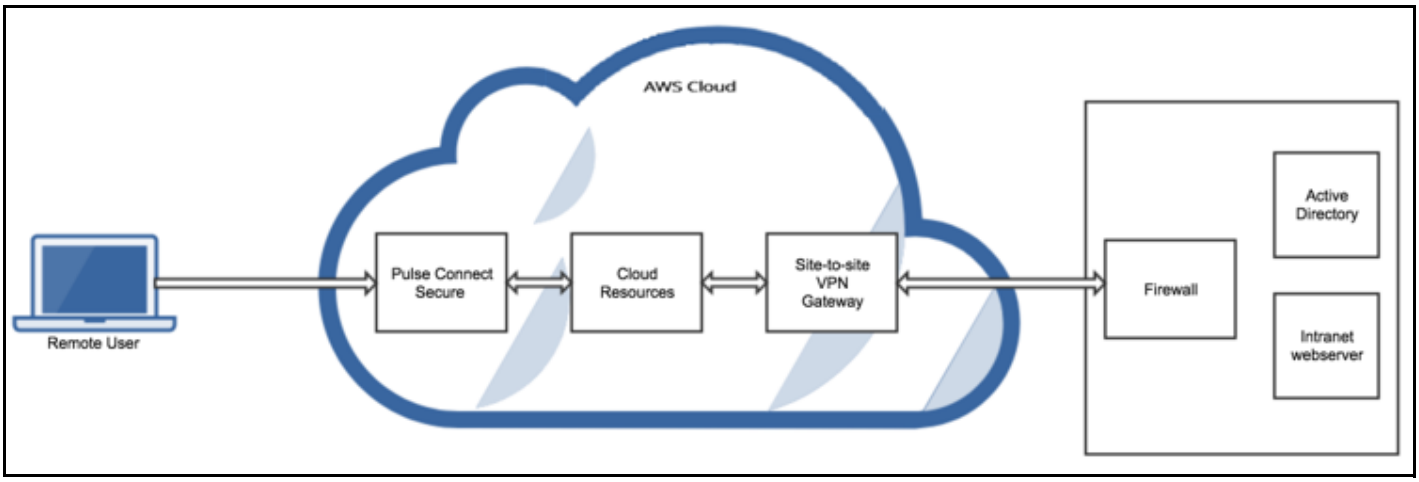
After successfully deploying PCS on AWS, go to the Outputs section and copy the Pulse External Interface details.

Figure 40 Pulse External Interface

Outputs

Key	Value	Description	Export Name
InternalAddress	Public IP address: 52.9.161.26 Private IP address: 10.20.1.148	PCS Internal Interface details	
ManagementAddress	Public IP address: 13.57.66.165 Private IP address: 10.20.3.211	PCS Management Interface details	
InstanceId	i-0b90b75a93e6a005e	Instance Id of newly created instance	
ExternalAddress	Public IP address: 52.8.243.247 Private IP address: 10.20.2.252	PCS Extenal Interface details	

Figure 41 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

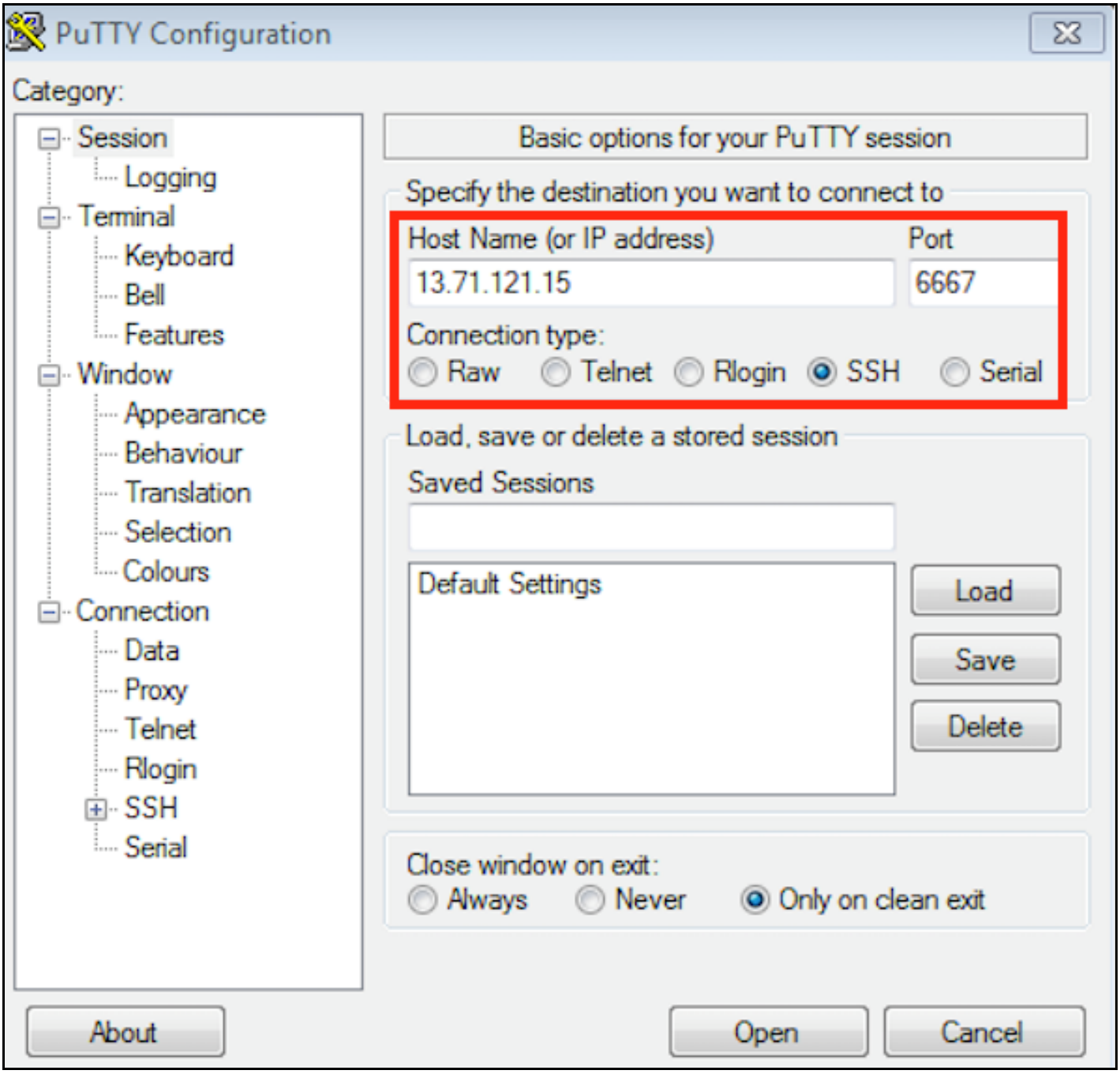
Execute the following command:

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

On Windows

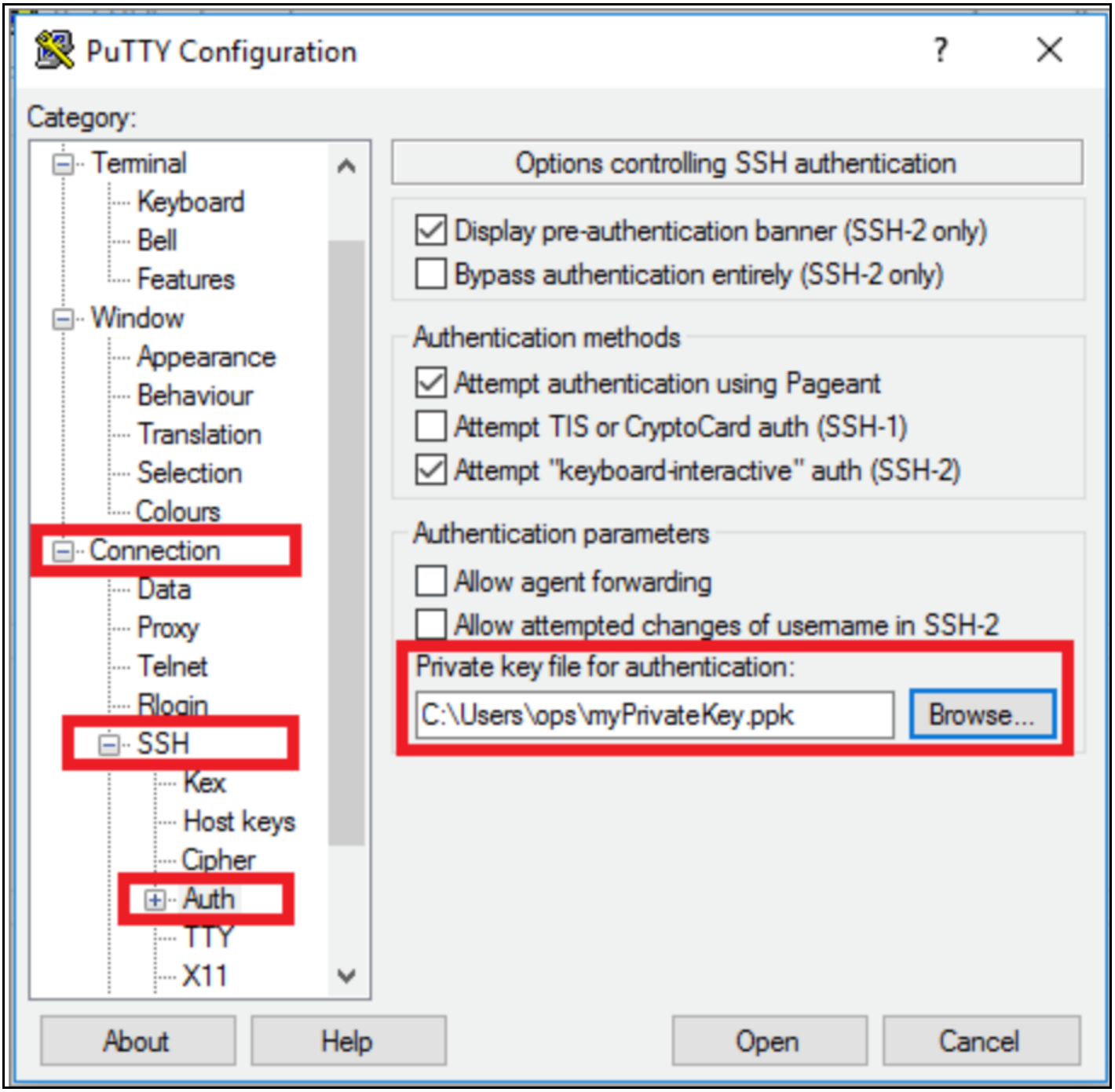
1. Launch the Putty terminal emulator.
2. In the Session category:
   - Enter the host name or IP address.
   - Enter the port number.
   - Select the connection type as SSH.

Figure 42 Putty Configuration – Basic Options



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

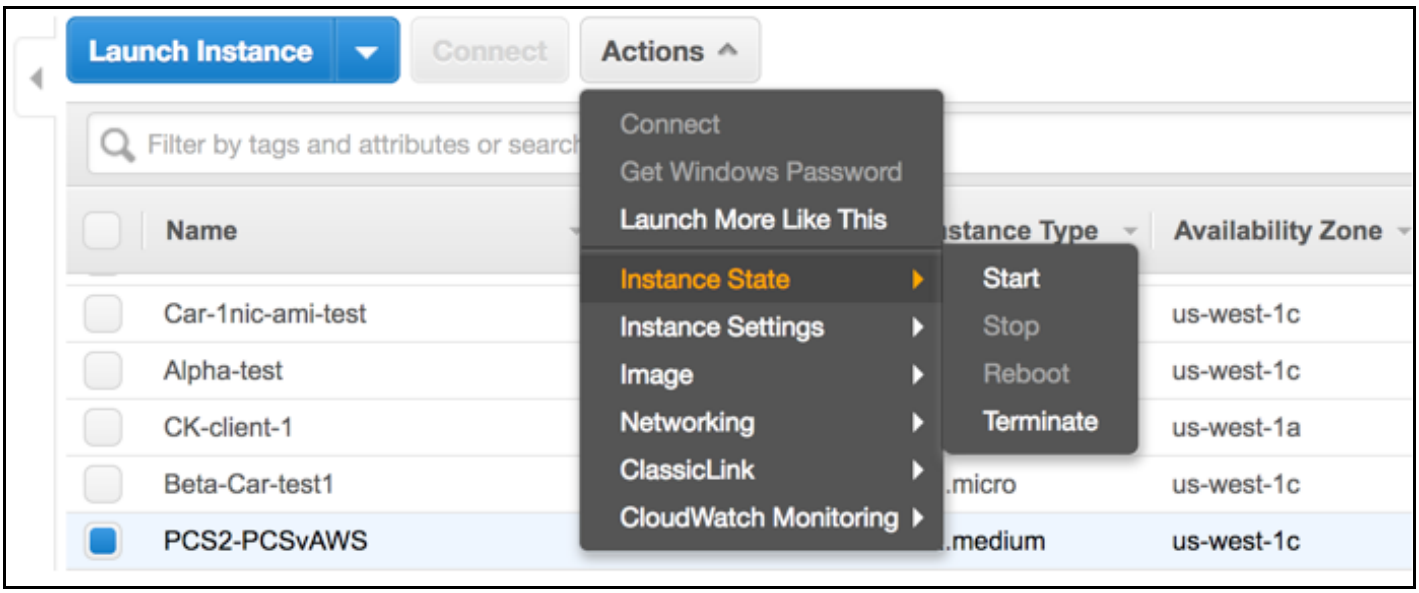
Figure 43 Putty Configuration – SSH Authentication



System Operations

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

Figure 44 System Operations



On the AWS portal, select **AWS Services > Launch Instance.** From the **Actions** menu, select **Instance State**.

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

Network Configuration

IP Address Assignment for Internal, External and Management Interfaces

Each interface in AWS can have private and public IP addresses. Sample CloudFormation 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 AWS can be seen at: https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-ip-addressing.html

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.

PCS external interface and PCS management interface have both Elastic and Private IP addresses.

Modifying Network Parameters After Deployment

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

Controlling the Selection of Internal, External and Management Interfaces

Sample CloudFormation template, provided by Pulse Secure, requests AWS fabric to create three Network Interfaces. While running this template, AWS 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 CloudFormation template?

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.

```
"EC2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": {"Ref": "PCSImageAMIId"},
    "KeyName": {"Ref": "KeyName"},
    "InstanceType": {"Ref": "InstanceType"},
    "NetworkInterfaces": [
      {"NetworkInterfaceId": {"Ref": "Eth0"}, "DeviceIndex": "0"},
      {"NetworkInterfaceId": {"Ref": "Eth1"}, "DeviceIndex": "1"},
      {"NetworkInterfaceId": {"Ref": "Eth2"}, "DeviceIndex": "2"}
    ],
    "Tags"     : [
      {"Key": "Name",
       "Value": {"Fn::Join": [ "-", [ { "Ref": "AWS::StackName" },
"PCSvAWS" ] ] }
      }
      ],
      "UserData": {"Fn::Base64": {"Fn::Join": ["", [{"Ref":
"PCSConfigData"}]]}}
  }
},
```

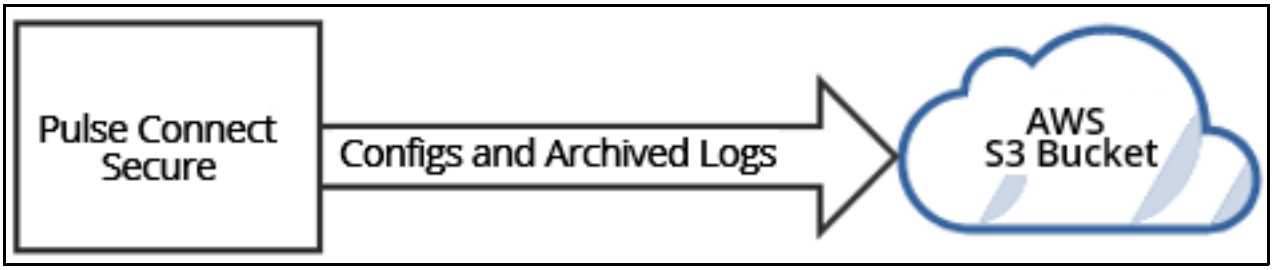
PCS converts eth0 to int0, eth1 to ext0 and eth2 to mgmt0. This means, the network interface with ID nic1 will be internal interface, nic2 will be external interface and nic3 will be management interface.

The below table depicts this scenario well:

Interface Name	PCS Interface	Network ID
eth0	internal interface	nic1
eth1	external interface	nic2
eth2	management interface	nic3

Backing up Configs and Archived Logs on S3 Bucket

Pulse Connect Secure supports pushing configs and archived logs to the servers that support SCP and FTP protocols. In the AWS deployment, Pulse Connect Secure now supports pushing configs and archived logs to the S3 bucket.



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 **AWS** option and configure S3 Bucket Name, AWS Access Key, AWS Secret Key, S3 Bucket Location and Destination Path Prefix.

Table 1 AWS Archive Settings

Parameter	Description
S3 Bucket Name	To create an S3 bucket: Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/. Select **Create bucket**. In the **Bucket name** field, type a unique DNS-compliant name for your new bucket. For more details about S3 bucket name, refer https://docs.aws.amazon.com/AmazonS3/latest/gsg/CreatingABucket.html
Region	S3 bucket location.

Parameter	Description
AWS Access Key	To create AWS Access Key and AWS Secret Key: Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/ In the navigation bar on the upper right, select your user name, and then select **My Security Credentials**. On the **AWS IAM Credentials** tab, in the Access keys for **CLI, SDK,** and **API access section,** select **Create access key.** Then select **Download .csv file** to save the access key ID and secret access key to a .csv file on your computer. When you create an access key, the key pair (access key ID and secret access key) is active by default, and you can use the pair right away. For more details, refer https://aws.amazon.com/premiumsupport/knowledge-center/create-access-key/
AWS Secret Key	See the procedure described for AWS Access Key. For more details, refer https://help.bittitan.com/hc/en-us/articles/115008255268-How-do-I-find-my-AWS-Access-Key-and-Secret-Access-Key-
Dest Path Prefix (Optional)	Path to copy files under S3 bucket.

Configuring Backup Configs and Archived Logs via REST

Setting AWS as Archive Logs Backup

REQUEST

```
PUT /api/v1/configuration/system/maintenance/archiving/settings HTTP/1.1
Content-Type: application/json
(
      "archive-path": "folder1/folder2",
      "directory": "ap-south-1",
      "method": "AWS",
      "Password-cleartext": "xkjdsklukjkwej",
      "server": "S3-server-storage-bucket",
      "user-name": "ADDDDDFVFFFQXXXXXA"
)
```

Mapping of keys in POST body:

archive-path	Destination path prefix
directory	Region

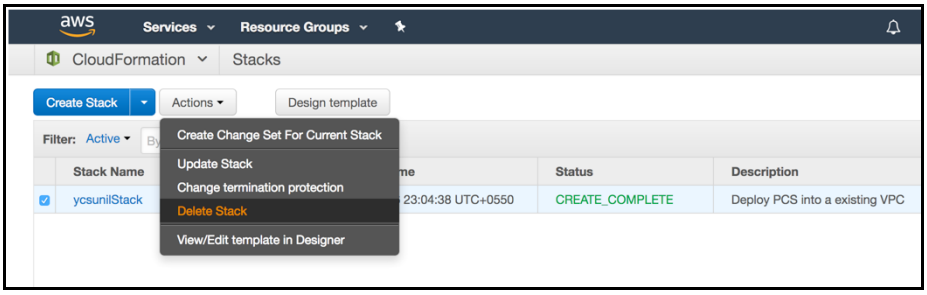
method	method (AWS)
Password-cleartext	AWS Secret key
server	S3 Bucket Name
user-name	AWS Access key

Decommissioning Pulse Connect Secure

To decommission Pulse Connect Secure, perform the following steps:

1. Select **AWS Services > CloudFormation.**
2. Click **Actions**. From the drop-down list displayed, select **Delete Stack.**

Figure 45 Delete Stack



Pricing

The cost of running this product is combination of License cost and AWS infrastructure cost. It will be very difficult to find out AWS infrastructure cost for this product as it may vary with Regions/Country/Time. Hence, we recommend using "AWS Calculator" which is available online to calculate the cost of running this product. https://calculator.s3.amazonaws.com/index.html

Here are resources that are created during deployment. Highlighted ones are chargeable in AWS.

Resources	Category	Chargeable
PCS VM (t2.medium / t2.xlarge / t2.2xlarge)	Compute	Yes
Virtual Private Cloud with four subnets	Networking	No
Three NICs named PCSInternalNIC, PCSExternalNIC and PCSManagementNIC	Networking	No
Three Elasti Public IPs for internal, external and management interfaces	Networking	Yes
Three Security Groups named SGInternal, SGExternal and SGManagement	Networking	No
Route table	Networking	No
PCS IMG file of size 40GB in S3 bucket	Storage	Yes
PCS Snapshot file of size 40GB in Elastic block store	Storage	Yes

Limitations

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

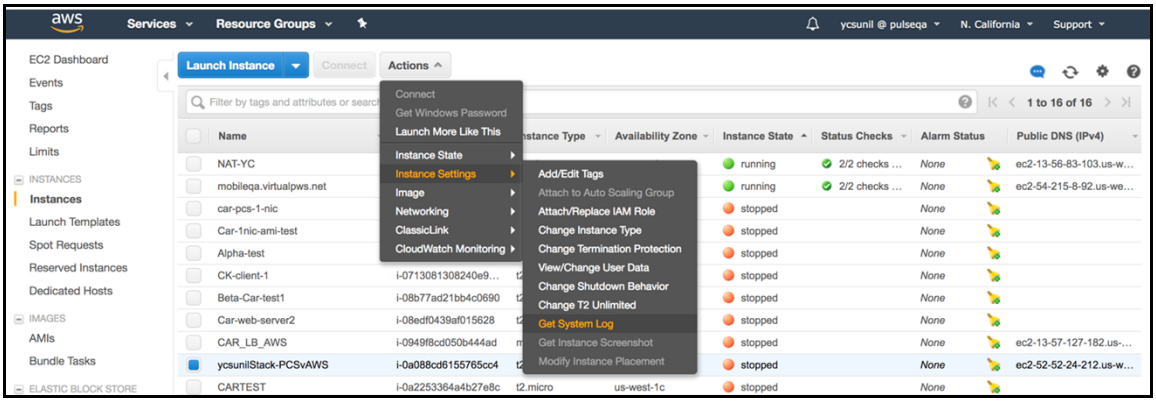
- IP address (private) of the interfaces should not be changed
- IPV6 is not supported

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:

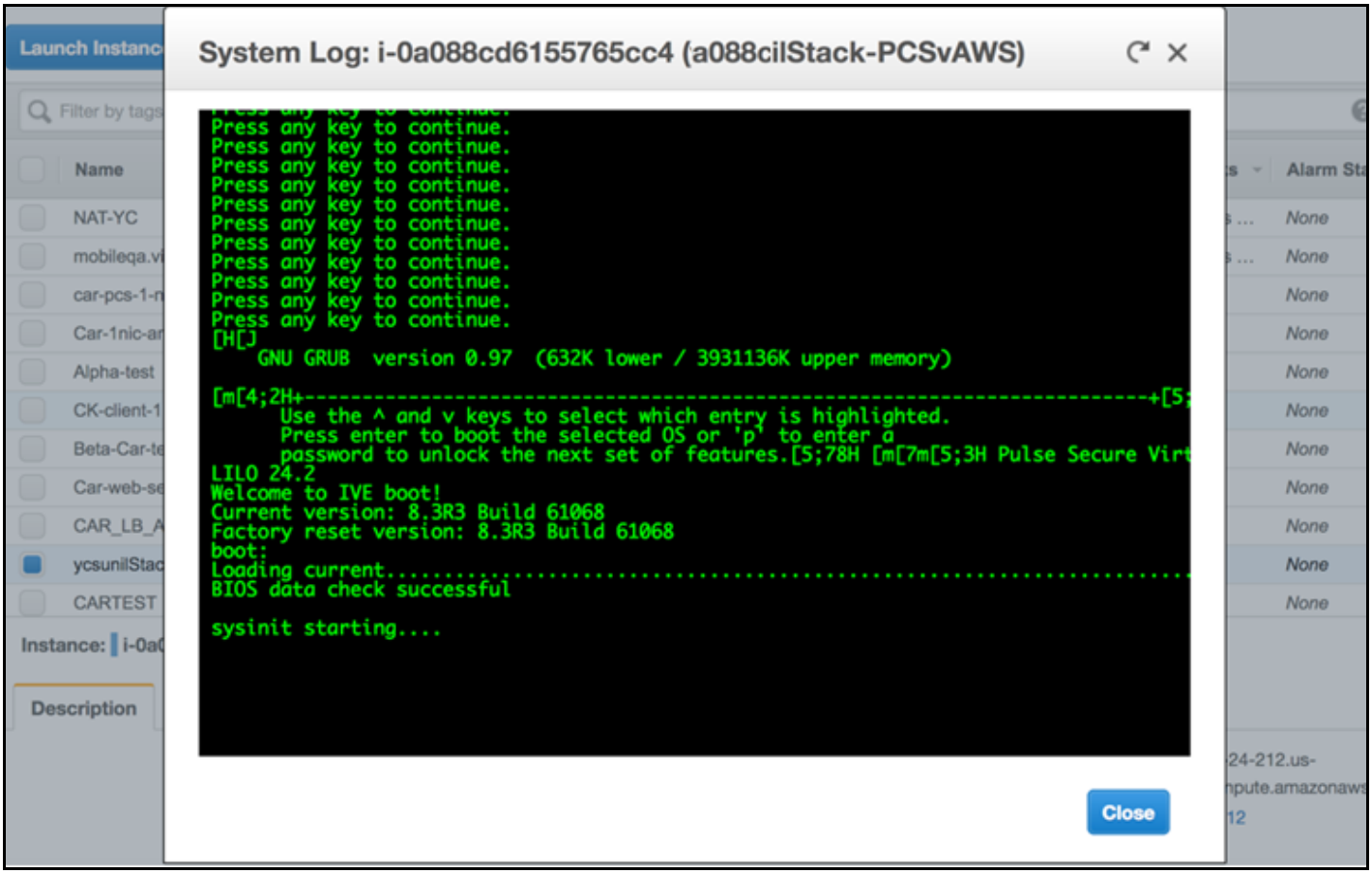
1. Select **AWS Services > Instances > Launch Instance.**
2. From the list displayed, select **Instance Settings > Get System Log.**

Figure 46 Boot Diagnostics



The system logs window is displayed.

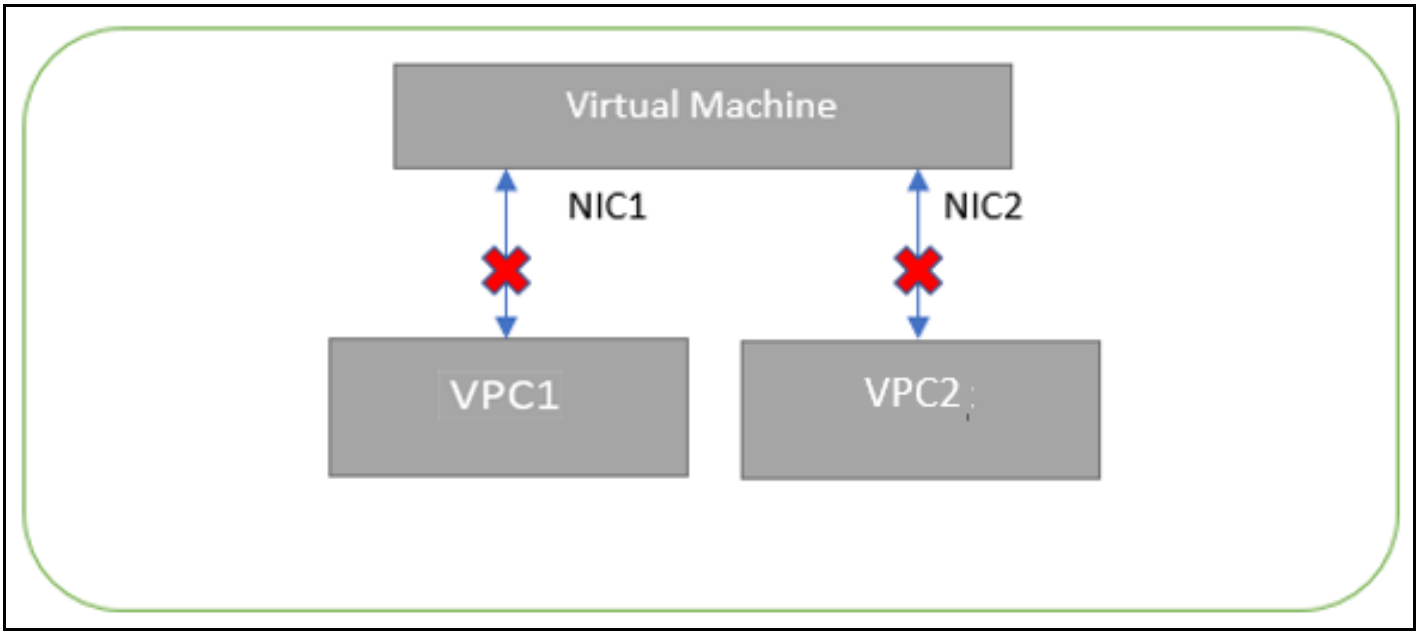
Figure 47 System Logs



Appendix A: Security Group (SG)

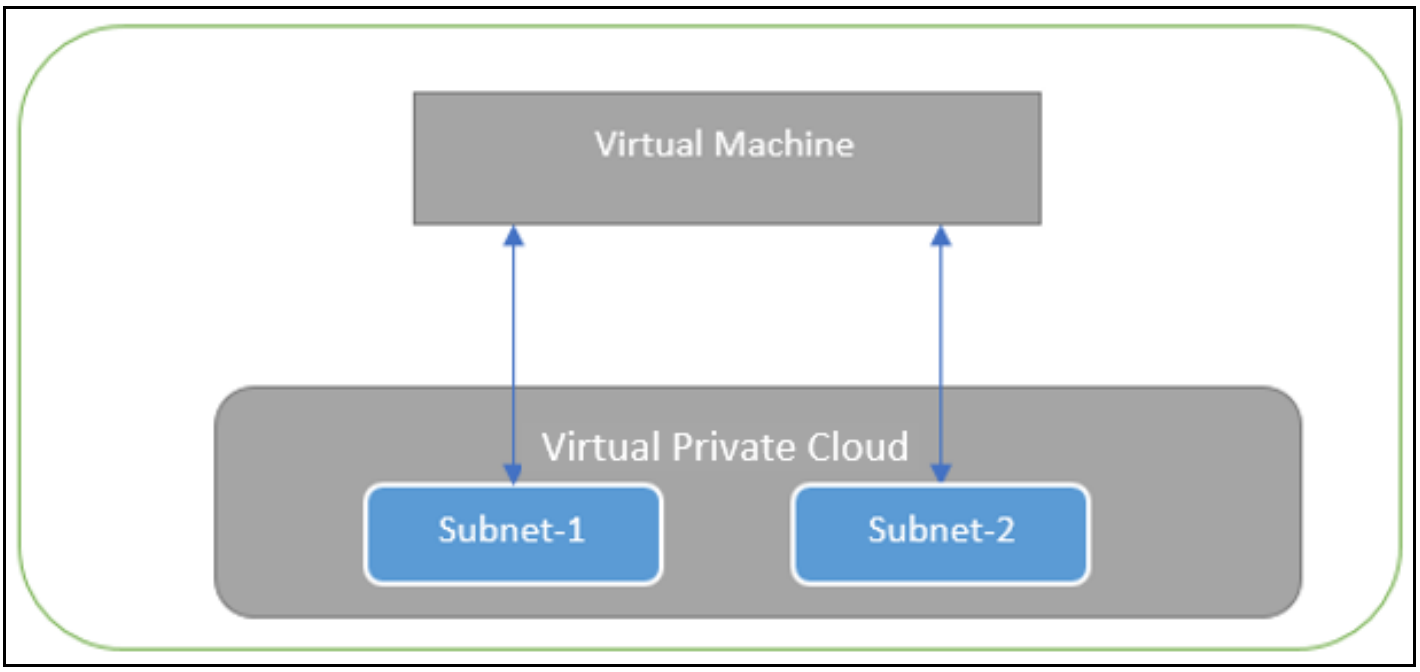
AWS has a limitation where virtual machine with multiple network interfaces cannot connect to different Virtual Private Cloud (VPCs). For example, a VM with two NICs, NIC1 and NIC2, will not be able to connect to VPC1 and VPC2 respectively.

Figure 48 Virtual Machine with two NICs Connecting to VPC1 and VPC2



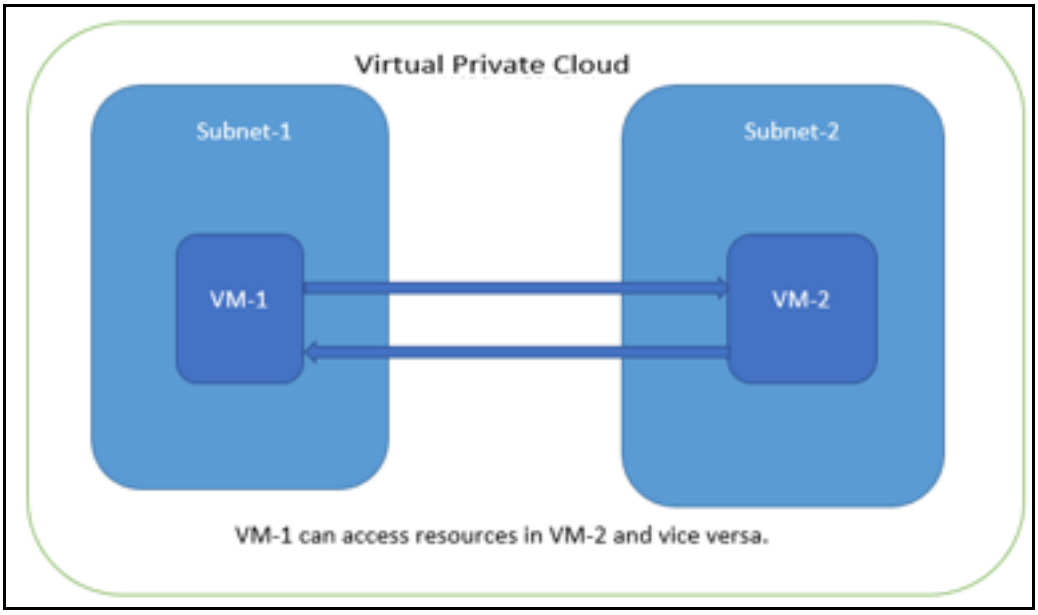
AWS supports a virtual machine with multiple NICs to connect to different Subnets under a same Virtual Private Cloud. For example, a VM with two NICs, NIC1 and NIC2, can connect to 'Subnet1' and 'Subnet2' where these subnets exist under a same Virtual Private Cloud respectively.

Figure 49 Virtual Machine with two NICs Connecting to Subnet1 and Subnet2



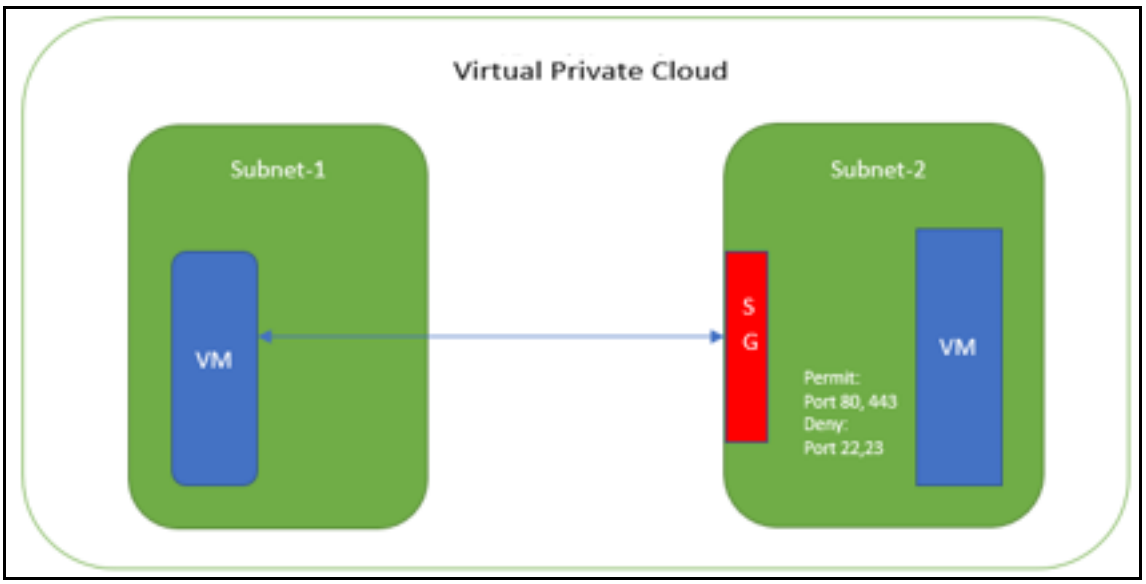
AWS provides isolation between different VPCs. But it does not provide the same kind of isolation when it comes to subnets in the same VPC. For example, consider a VPC has two subnets, Subnet1 and Subnet2. And consider two VMs, VM-1 and VM-2, which are connected to Subnet1 and Subnet2 respectively. In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

Figure 50 Virtual Machine VM-1 can Access Resources in VM-2 and Vice Vers



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 “Security Group” provided by AWS.

Figure 51 Traffic Filtering by AWS Support Group



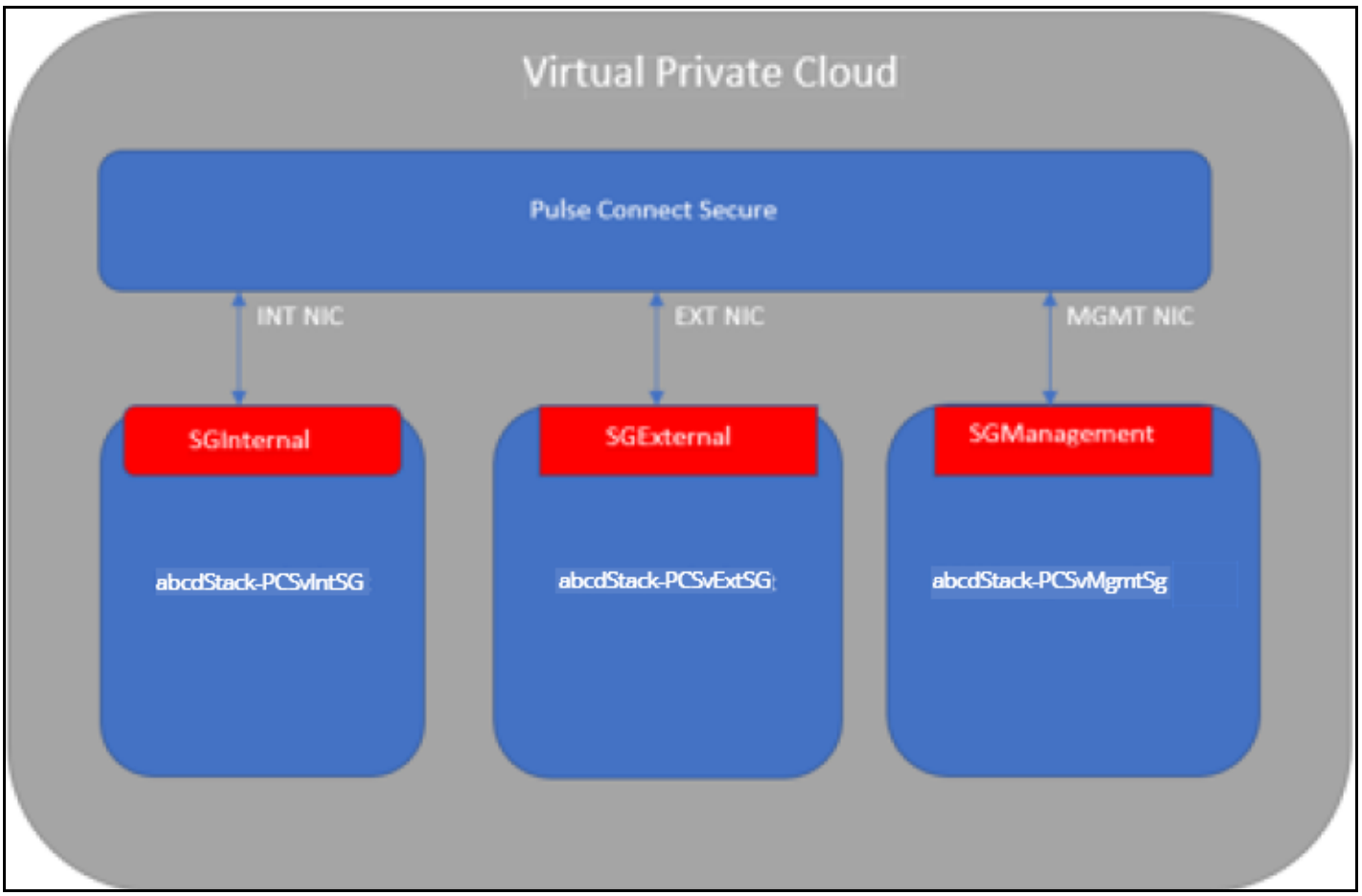
Pulse Connect Secure, when provisioned through the CloudFormation template provided by Pulse Secure, creates four subnets under a virtual private cloud named “PCSVirtualNetwork”. The four Subnets are:

- PCSInternalSubnet
- PCSExternalSubnet
- PCSManagementSubnet
- PCSTunnelVPNPoolSubnet

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

- SGExternalSubnet
- SGInternalSubnet
- SGManagementSubnet

Figure 52 SG External, Internal and Management Subnets



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

1. The list of SG Inbound/Outbound rules created **"Stack-PCSvExtSG"** are:

Figure 53 Stack-PCSvExtSG - Inbound Rules

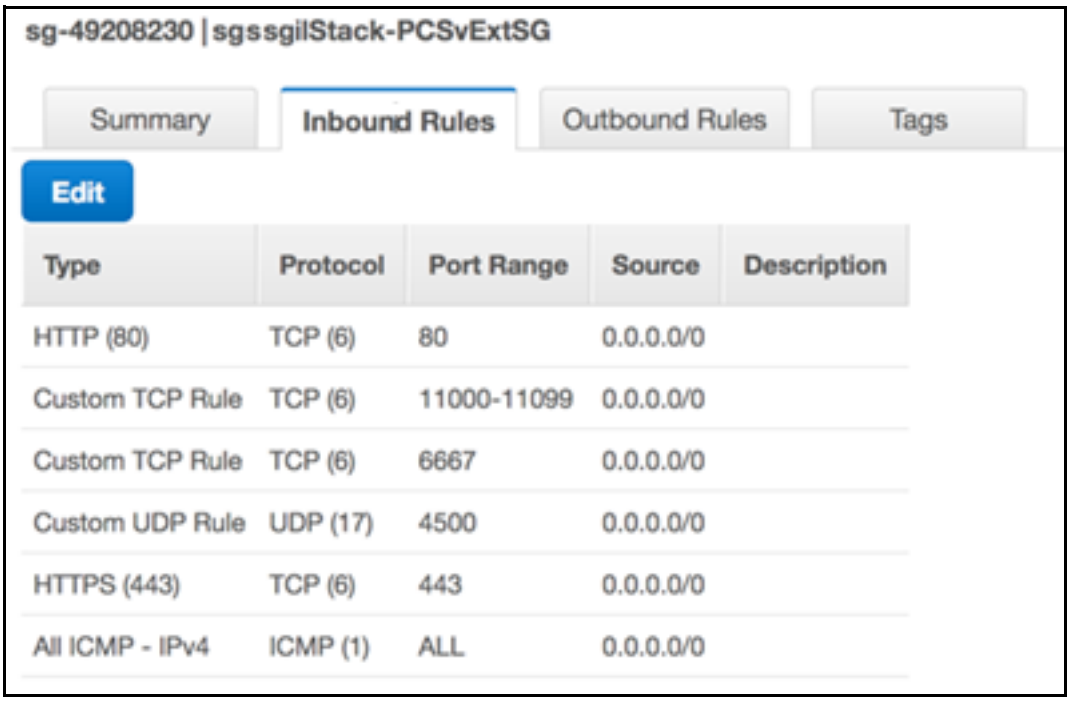
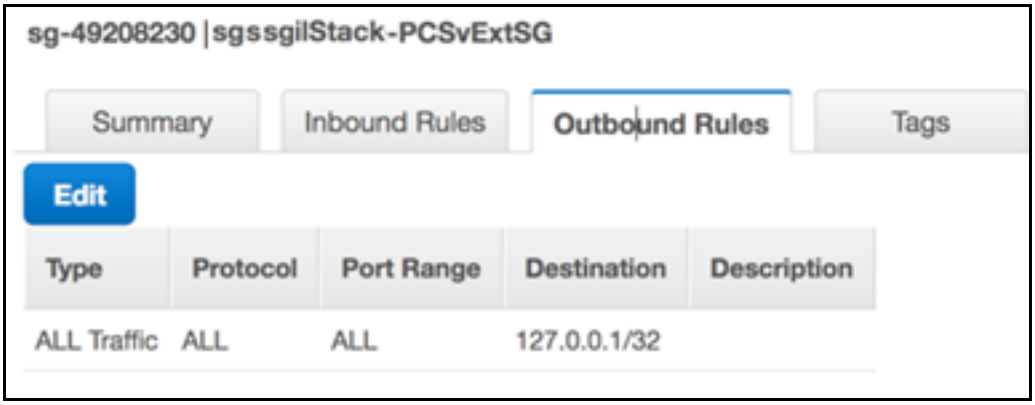


Figure 54 Stack-PCSvExtSG - Outbound Rules



2. The list of SG Inbound/Outbound rules created **"Stack-PCSvIntSG"** are:

Figure 55 Stack-PCSvIntSG - Inbound Rules

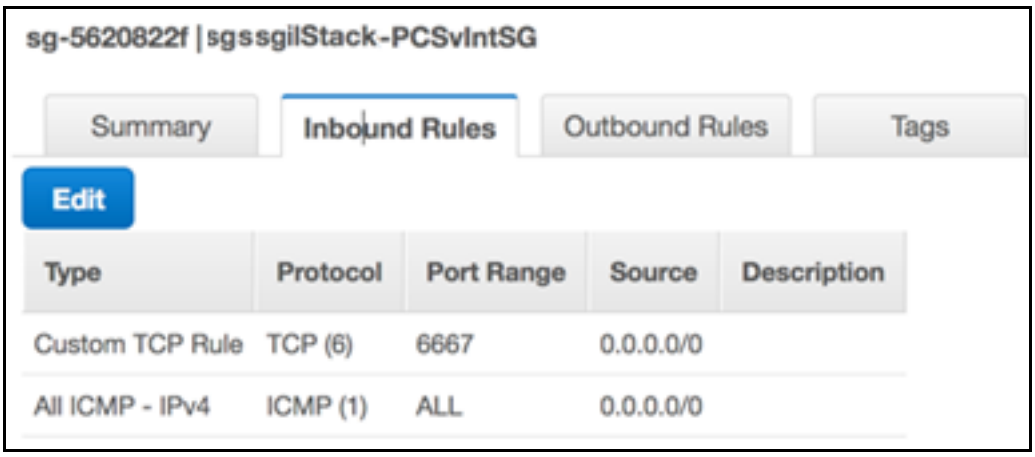
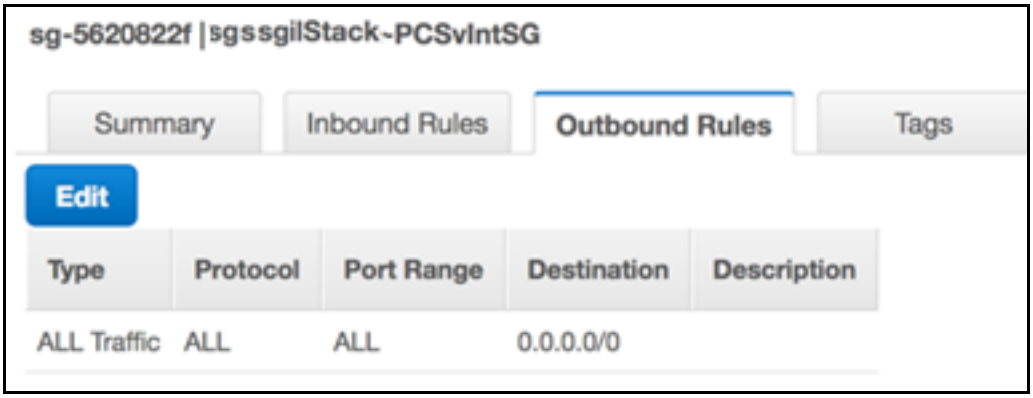


Figure 56 Stack-PCSvIntSG - Outbound Rules



3. The list of SG Inbound/Outbound rules created **"Stack-PCSvMgmtSG"** are:

Figure 57 Stack-PCSvMgmtSG - Inbound Rules

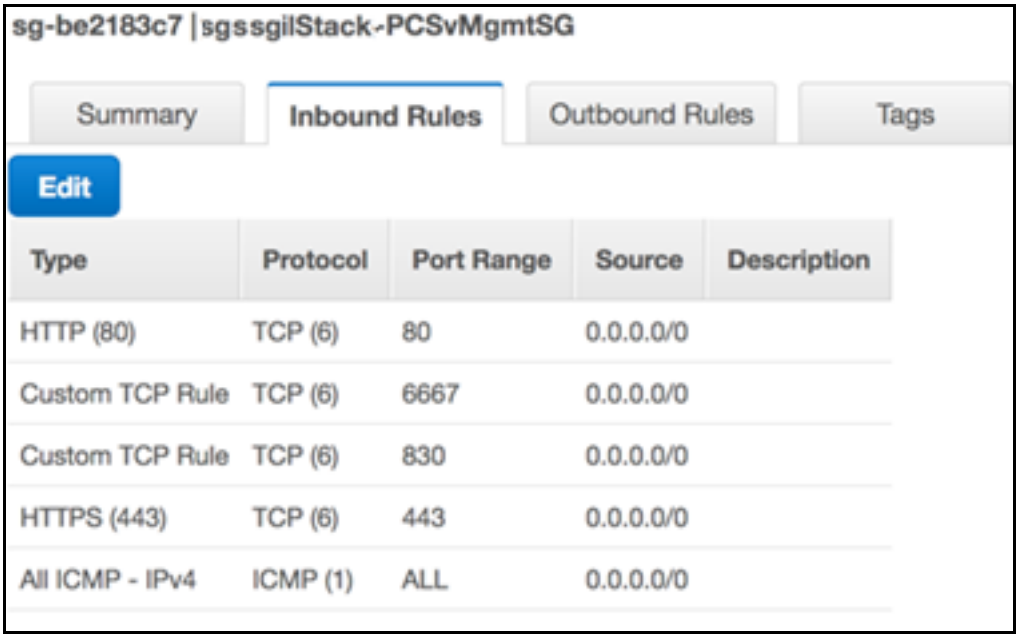
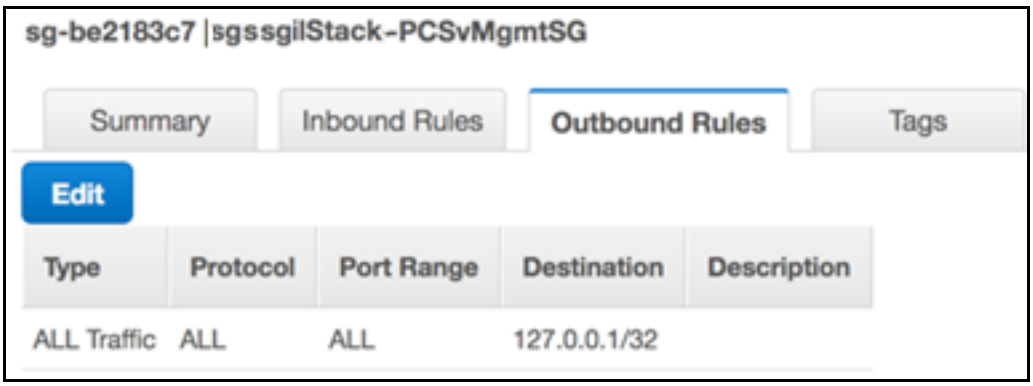


Figure 58 Stack-PCSvMgmtSG - Outbound Rules



Appendix B: Pulse Connect Secure CloudFormation Template

Pulse Secure provides sample CloudFormation template files to deploy the Pulse Connect Secure Virtual Appliance on AWS. Users can modify this to make it suitable for their need. Visit Amazon marketplace and download the pulsesecure-pcs-3-nics.zip file, and unzip it to get **pulsesecure-pcs-3-nics-new-network.json.**

This template creates a new PCS with 3 NICs, VPC, four subnets, security group 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 AWS. It contains parameter name, its default value and the mouse-over help text that is displayed when mouse is placed over the parameter in AWS Web portal. The parameters defined here are displayed in the Custom Deployment page of AWS portal.
Resources	This section defines resource types that are deployed or updated in a resource group.
Outputs	This section defines the public IP address, private IP address and primary private IP address returned after successful deployment of PCS on AWS.

Parameters

Key Name: This is the name of the PCS Storage Account where the PCS VA image (.ami file) is stored.

```
"Parameters" : {

  "KeyName": {
      "Type": "AWS::EC2::KeyPair::KeyName",
      "Default": "",
      "AllowedPattern" : "[-_ a-zA-Z0-9]*",
      "Description": "Name of an existing EC2 KeyPair. Your PCS will launch with this KeyPair.",
      "ConstraintDescription": "Must be the name of an existing EC2 KeyPair."
    },
```

PCS Image AMI ID: This is the ID of the uploaded AMI file.

```
"PCSImageAMIId" : {
  "Type" : "String",
  "Description" : "AMI ID of your existing PCS image"
},
```

Instance Type: This specifies the size of the instance – t2.medium or t2.large

```
"InstanceType": {
  "Description": "Select PCS instance type",
  "Type": "String",
  "Default": "t2.medium",
  "AllowedValues": [
      "t2.medium",
      "t2.large"
  ],
  "ConstraintDescription": "Must be an allowed EC2 instance type."
},
```

PCS Config Data: 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 AWS cloud or in the corporate network which is accessible for Pulse Connect Secure through site-to-site VPN between AWS and the corporate data center.

Pulse Connect Secure accepts the following parameters as provisioning parameters:

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

For details about these parameters, see “Pulse Connect Secure Provisioning Parameters” on page 34.

```
"<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></config-download-url><config-data></config-data><auth-code-license></auth-code-license><enable-license-server>n</enable-license-server><accept-license-agreement>n</accept-license-agreement><enable-rest>n</enable-rest></pulse-config>"
```

VPC CIDR: It is a valid CIDR range of the form x.x.x.x/x for entire VPC.

```
"VPCCIDR": {
  "Description": "CIDR block for entire VPC.",
  "Type": "String",
  "Default": "10.20.0.0/16",
  "AllowedPattern":
  "^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\\/([0-9]|[1-2][0-9]|3[0-2]))$",
  "ConstraintDescription": "Must be a valid CIDR range of the form x.x.x.x/x."
},
```

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

```
"InternalSubnetCIDR": {
  "Description": "PCS internal interface connects to this subnet",
  "Type": "String",
  "Default": "10.20.1.0/24",
  "AllowedPattern":
  "^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\\/([0-9]|[1-2][0-9]|3[0-2]))$",
  "ConstraintDescription": "CIDR block parameter must be in the form x.x.x.x/x"
},
```

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

```
"ExternalSubnetCIDR": {
  "Description": "PCS external interface connects to this subnet",
  "Type": "String",
  "Default": "10.20.2.0/24",
  "AllowedPattern":
  "^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\\/([0-9]|[1-2][0-9]|3[0-2]))$",
  "ConstraintDescription": "CIDR block parameter must be in the form x.x.x.x/x"
},
```

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

```
"ManagementSubnetCIDR": {
  "Description": "PCS management interface connects to this subnet",
  "Type": "String",
  "Default": "10.20.3.0/24",
  "AllowedPattern":
  "^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\\/([0-9]|[1-2][0-9]|3[0-2]))$",
  "ConstraintDescription": "CIDR block parameter must be in the form x.x.x.x/x"
},
```

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

```
"TunnelSubnetCIDR": {
  "Description": "For L3 VPN connections PCS hands over IP to the clients from this subnet",
  "Type": "String",
  "Default": "10.20.4.0/24",
  "AllowedPattern":
  "^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\\/([0-9]|[1-2][0-9]|3[0-2]))$",
  "ConstraintDescription": "CIDR block parameter must be in the form x.x.x.x/x"
}
```

Resources

VPC:

```
"VPC" : {
    "Type" : "AWS::EC2::VPC",
```

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

```
"IntSubnet" : {
  "Type" : "AWS::EC2::Subnet",
```

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

```
 "ExtSubnet" : {
   "Type" : "AWS::EC2::Subnet",
```

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

```
"MgmtSubnet" : {
  "Type" : "AWS::EC2::Subnet",
```

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

```
"TunnelSubnet" : {
  "Type" : "AWS::EC2::Subnet",
```

InternetGateway:

```
"InternetGateway" : {
  "Type" : "AWS::EC2::InternetGateway",
```

AttachGateway:

```
"AttachGateway" : {
   "Type" : "AWS::EC2::VPCGatewayAttachment",
```

PublicSubnetRouteTable:

```
"PublicSubnetRouteTable" : {
  "Type" : "AWS::EC2::RouteTable",
```

PublicSubnetRoute:

```
"PublicSubnetRoute" : {
  "Type" : "AWS::EC2::Route",
```

ExtSubnetRouteTableAssociation:

```
"ExtSubnetRouteTableAssociation" : {
  "Type" : "AWS::EC2::SubnetRouteTableAssociation",
```

MgmtSubnetRouteTableAssociation:

```
"MgmtSubnetRouteTableAssociation" : {
  "Type" : "AWS::EC2::SubnetRouteTableAssociation",
```

EIP1:

```
"EIP1" :  {
  "Type" : "AWS::EC2::EIP",
```

EIPAssoc1:

```
"EIPAssoc1" : {
  "Type" : "AWS::EC2::EIPAssociation",
```

EIP2:

```
"EIP2" :  {
  "Type" : "AWS::EC2::EIP",
```

EIPAssoc2:

```
"EIPAssoc2" : {
  "Type" : "AWS::EC2::EIPAssociation",
```

PCSvExternalSecurityGroup:

```
"PCSvExternalSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
```

PCSvInternalSecurityGroup:

```
"PCSvInternalSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
```

PCSvManagementSecurityGroup:

```
"PCSvManagementSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
```

EC2Instance:

```
"EC2Instance" : {
  "Type" : "AWS::EC2::Instance",
```

Eth0:

```
"Eth0" : {
  "Type" : "AWS::EC2::NetworkInterface",
```

Eth1:

```
"Eth1" : {
  "Type" : "AWS::EC2::NetworkInterface",
```

Eth2:

```
"Eth2" : {
  "Type" : "AWS::EC2::NetworkInterface",
```

Outputs

The Outputs section defines the public IP address, private IP address and primary private IP address that is displayed on successful deployment of PCS on AWS.

```
"Outputs" : {
  "InstanceId" : {
    "Value" : { "Ref" : "EC2Instance" },
    "Description" : "Instance Id of newly created instance"
  },
  "ManagementAddress" : {
    "Value" : { "Fn::Join" : [" ", [ "Public IP address:", { "Ref" : "EIP2" }, "Private IP address:", { "Fn::GetAtt" : ["Eth2", "PrimaryPrivateIpAddress"]
    }]]},
    "Description" : "PCS Management Interface details"
  },
  "ExternalAddress" : {
    "Value" : { "Fn::Join" : [" ", [ "Public IP address:", { "Ref" : "EIP1" }, "Private IP address:", { "Fn::GetAtt" : ["Eth1", "PrimaryPrivateIpAddress"]
    }]]},
    "Description" : "PCS Extenal Interface details"
  },
  "InternalAddress" : {
    "Value" : { "Fn::Join" : [" ", [ "Private IP address:", { "Fn::GetAtt" : ["Eth0", "PrimaryPrivateIpAddress"] }]]},
    "Description" : "PCS Internal Interface details"
  }
}
```

Appendix C: Pulse Connect Secure CloudFormation Template for an Existing Virtual Private Cloud

Pulse Secure provides sample CloudFormation template files to deploy Pulse Connect Secure Virtual Appliance on AWS. Users can modify this to make it suitable for their need. Visit Amazon marketplace and download the pulsesecure-pcs-3-nics.zip file, and unzip it to get **pulsesecure-pcs-3-nics-existing-vpc.json.**

This template creates a new PCS with 3 NICs, VPC, four subnets, security group 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	Resources	Outputs
This section defines the parameters used for deploying PCS on AWS. It contains parameter name, its default value and the mouse-over help text that is displayed when mouse is placed over the parameter in AWS Web portal. The parameters defined here are displayed in the Custom Deployment page of AWS portal.	This section defines resource types that are deployed or updated in a resource group.	This section defines the public IP address and FQDN returned after successful deployment of PCS on AWS.

Parameters

Key Name: This is the name of the PCS Storage Account where the PCS VA image (.ami file) is stored.

```
"Parameters" : {

  "KeyName": {
      "Type": "AWS::EC2::KeyPair::KeyName",
      "Default": "",
      "AllowedPattern" : "[-_ a-zA-Z0-9]*",
      "Description": "Name of an existing EC2 KeyPair. Your PCS will launch with this KeyPair.",
      "ConstraintDescription": "Must be the name of an existing EC2 KeyPair."
    },
```

PCS Image AMI ID: This is the ID of the uploaded AMI file.

```
"Parameters" : {

 "KeyName": {
     "Type": "AWS::EC2::KeyPair::KeyName",
     "Default": "",
     "AllowedPattern" : "[-_ a-zA-Z0-9]*",
     "Description": "Name of an existing EC2 KeyPair. Your PCS will launch with this KeyPair.",
     "ConstraintDescription": "Must be the name of an existing EC2 KeyPair."
  },
```

Instance Type: This specifies the size of the instance – t2.medium or t2.large

```
"PCSImageAMIId" : {
  "Type" : "String",
  "Description" : "AMI ID of your existing PCS image"
},
```

PCS Config Data: 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 AWS cloud or in the corporate network which is accessible for Pulse Connect Secure through site-to-site VPN between AWS and the corporate data center.

Pulse Connect Secure accepts the following parameters as provisioning parameters:

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

For details about these parameters, see “Pulse Connect Secure Provisioning Parameters” on page 34.

VPCID: This is the ID of the existing VPC.

```
"InstanceType": {
  "Description": "Select PCS instance type",
  "Type": "String",
  "Default": "t2.medium",
  "AllowedValues": [
      "t2.medium",
      "t2.large"
  ],
  "ConstraintDescription": "Must be an allowed EC2 instance type."
},
```

SubnetIntID: This is the ID of the subnet to which PCS Internal interface connects.

```
"<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>pssecure.net</dns-domain><admin-username>admin</admin-username><admin-password>password</admin-password><cert-common-name>va1.pssecure.net</cert-common-name><cert-random-text>fdsfpisonvsfnms</cert-random-text><cert-organisation>Pssecure Org</cert-organisation><config-download-url></config-download-url><config-data></config-data><auth-code-license></auth-code-license><enable-license-server>n</enable-license-server><accept-license-agreement>n</accept-license-agreement><enable-rest>n</enable-rest></pulse-config>"
```

SubnetExtId: This is the ID of the subnet to which PCS External interface connects.

```
"VpcId" : {
  "Type" : "String",
  "Description" : "ID of existing VPC"
},
```

SubnetMgmtId: This is the ID of the subnet to which PCS Management interface connects.

```
"SubnetIntId" : {
  "Type" : "String",
  "Description" : "ID of the subnet where PCS internal interface connects"
},
```

Resources

EIP1:

```
"EIP1" :  {
  "Type" : "AWS::EC2::EIP",
```

EIPAssoc1:

```
"EIPAssoc1" : {
  "Type" : "AWS::EC2::EIPAssociation",
```

EIP2:

```
"EIP2" :  {
  "Type" : "AWS::EC2::EIP",
```

EIPAssoc2:

```
"EIPAssoc2" : {
  "Type" : "AWS::EC2::EIPAssociation",
```

PCSvExternalSecurityGroup:

```
"PCSvExternalSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
```

PCSvInternalSecurityGroup:

```
 "PCSvInternalSecurityGroup": {
   "Type": "AWS::EC2::SecurityGroup",
```

PCSvManagementSecurityGroup:

```
"PCSvManagementSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
```

EC2Instance:

```
 "EC2Instance" : {
   "Type" : "AWS::EC2::Instance",
```

Eth0:

```
 "Eth0" : {
   "Type" : "AWS::EC2::NetworkInterface",
```

Eth1:

```
"Eth1" : {
  "Type" : "AWS::EC2::NetworkInterface",
```

Eth2:

```
"Eth2" : {
  "Type" : "AWS::EC2::NetworkInterface",
```

Outputs

The Outputs section defines the public IP address, private IP address and primary private IP address that is displayed on successful deployment of PCS on AWS.

```
"Outputs" : {
  "InstanceId" : {
    "Value" : { "Ref" : "EC2Instance" },
    "Description" : "Instance Id of newly created instance"
  },
  "ManagementAddress" : {
    "Value" : { "Fn::Join" : [" ", [ "Public IP address:", { "Ref" : "EIP2" }, "Private IP address:", { "Fn::GetAtt" : ["Eth2", "PrimaryPrivateIpAddress"]
    }]]},
    "Description" : "PCS Management Interface details"
  },
  "ExternalAddress" : {
    "Value" : { "Fn::Join" : [" ", [ "Public IP address:", { "Ref" : "EIP1" }, "Private IP address:", { "Fn::GetAtt" : ["Eth1", "PrimaryPrivateIpAddress"]
    }]]},
    "Description" : "PCS Extenal Interface details"
  },
  "InternalAddress" : {
    "Value" : { "Fn::Join" : [" ", [ "Private IP address:", { "Fn::GetAtt" : ["Eth0", "PrimaryPrivateIpAddress"] }]]},
    "Description" : "PCS Internal Interface details"
  }
}
```

References

AWS documentation: https://aws.amazon.com/documentation/

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.