



Pulse Connect Secure Virtual Appliance on Amazon Web Services Deployment Guide

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 Amazon Web Services - 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://www.pulsesecure.net>. 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
2.0 March 2019	Added deploying on AWS Marketplace	Added Deploying PCS on AWS Marketplace
1.0, September 2018	None	Document has no changes from the previous release

Table of Contents

Revision History	3
Overview	8
About This Guide	8
Assumptions	8
Pulse Connect Secure in AWS Marketplace	8
Prerequisites and System Requirements on AWS Marketplace	8
Deploying Pulse Connect Secure on AWS Marketplace	9
Select Template	10
Specify Details	11
Review	12
Pulse Connect Secure on Amazon Web Services	13
Prerequisites and System Requirements on AWS	13
Deploying Pulse Connect Secure on Amazon Web Services	13
Supported Platform Systems	14
Steps to Deploy Pulse Connect Secure on AWS	14
Registering the AMI	14
Prerequisites	14
Deploying Pulse Connect Secure on AWS using AWS Portal	15
Deploying PCS on New Virtual Private Cloud	16
Deployment on VM with Three NIC Cards	16
Deployment on VM with Two NIC Cards	18
Deploying PCS on an Existing Virtual Private Cloud	20
Deployment on VM with Three NIC Cards	20
Deployment on VM with Two NIC Cards	22
Deploying PCS as a License Server	24
Deploying PCS Active-Active Cluster using Virtual Traffic Manager in AWS	25
Deploying Two PCS EC2 instances Using CloudFormation Template	26
Forming the Active-Active Cluster	26
Deploying Virtual Traffic Manager EC2 Instance in the External Subnet of PCS in AWS	27
Setting Up and Configuring vTM for External Users	28
Pulse Connect Secure Provisioning Parameters	32
Provisioning Pulse Connect Secure with Predefined Configuration	33
Configuring Licenses on the Pulse Connect Secure Appliance	34
Pulse License Server in Corporate Network	34
Pulse License Server in Cloud Network	34
Adding Authentication Code in PCS Admin Console	35
Including Authentication Code in CloudFormation Template	35
Accessing the Pulse Connect Secure Virtual Appliance	36
Accessing the Pulse Connect Secure Virtual Appliance as an Administrator	36
Accessing the Pulse Connect Secure Virtual Appliance as an End User	36
Accessing the Pulse Connect Secure Virtual Appliance using SSH Console	37

On Linux and Mac OSX.....	37
On Windows	37
System Operations	39
Network Configuration	39
IP Address Assignment for Internal, External and Management Interfaces.....	39
IP Addressing Modes.....	39
Modifying Network Parameters After Deployment	39
Controlling the Selection of Internal, External and Management Interfaces	40
Decommissioning Pulse Connect Secure	41
Pricing.....	41
Limitations	41
Troubleshooting.....	42
Frequently Asked Questions	43
FAQ1: Packets transmitted from PCS Internal Interface are getting dropped by AWS Virtual Gateway in L3 traffic.....	43
Appendix A: Security Group (SG).....	46
Appendix B: Pulse Connect Secure CloudFormation Template	50
Parameters	50
Resources.....	52
Outputs.....	54
Appendix C: Pulse Connect Secure CloudFormation Template for an Existing Virtual Private Cloud	55
Parameters	55
Resources.....	57
Outputs.....	58
References.....	59
Requesting Technical Support.....	59

List of Figures

Figure 1: Pulse Connect Secure on AWS	14
Figure 2: Create New Stack.....	16
Figure 3: Upload Template	16
Figure 4: Specify Details for New Virtual Private Cloud.....	17
Figure 5: New VPC.....	18
Figure 6: Create New Stack.....	18
Figure 7: Upload Template	18
Figure 8: Specify Details for New Virtual Private Cloud.....	19
Figure 9: New VPC.....	20
Figure 10: Create New Stack	20
Figure 11: Upload Template.....	21
Figure 12: Specify Details for Existing Virtual Private Cloud	21
Figure 13: Create New Stack	22
Figure 14: Upload Template.....	22
Figure 15: Specify Details for Existing Virtual Private Cloud	23
Figure 16: Create New Stack	24
Figure 17: Upload Template.....	24
Figure 18: Specify Details for Existing Virtual Private Cloud	25
Figure 19: Deploying PCS A-A Cluster Topology Diagram	25
Figure 20: PCS A-A Cluster Status	26
Figure 21: AWS Marketplace > Pulse Secure vTM	27
Figure 22: vTM Editions Available in AWS Marketplace.....	27
Figure 23: 1-Click Launch Tab	28
Figure 24: Pulse Secure vTM Login Page.....	28
Figure 25: Create Traffic Pool.....	29
Figure 26: SSL and UDP Pools	29
Figure 27: Session Persistency Class.....	30
Figure 28: Create Virtual Server.....	30
Figure 29: Virtual Servers to Handle SSL and UDP Traffic.....	31
Figure 30: Pulse Secure vTM Home Page Showing Services and Event Logs.....	31
Figure 31: Pulse Configuration Server in Corporate Network	33
Figure 32: Pulse License Server in a Corporate Network.....	34
Figure 33: Pulse License Server in Cloud Network	34
Figure 34: Enter Authentication Code.....	35
Figure 35: Accessing PCS Virtual Appliance	36
Figure 36: Pulse External Interface	36
Figure 37: Resource in Corporate Network.....	37
Figure 38: Putty Configuration – Basic Options	38

Figure 39: Putty Configuration – SSH Authentication.....	38
Figure 40: System Operations.....	39
Figure 41: Delete Stack.....	41
Figure 42: Boot Diagnostics.....	42
Figure 43: System Logs.....	42
Figure 44: Topology diagram.....	43
Figure 45: Subnets.....	43
Figure 46: Route Table.....	44
Figure 47: Routes.....	44
Figure 48: Route Table.....	45
Figure 49: Routes.....	45
Figure 50: Virtual Machine with two NICs Connecting to VPC1 and VPC2	46
Figure 51: Virtual Machine with two NICs Connecting to Subnet1 and Subnet2.....	46
Figure 52: Virtual Machine VM-1 can Access Resources in VM-2 and Vice Versa.....	46
Figure 53: Traffic Filtering by AWS Support Group	47
Figure 54: SG External, Internal and Management Subnets.....	47
Figure 55: Stack-PCSVExtSG - Inbound Rules	48
Figure 56: Stack-PCSVExtSG - Outbound Rules.....	48
Figure 57: Stack-PCSVIntSG - Inbound Rules.....	48
Figure 58: Stack-PCSVIntSG - Outbound Rules	49
Figure 59: Stack-PCSVMgmtSG - Inbound Rules.....	49
Figure 60: Stack-PCSVMgmtSG - Outbound Rules	49

Overview

About This Guide

This guide helps in deploying the Pulse Connect Secure Virtual 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.

Pulse Connect Secure in AWS Marketplace

Beginning 9.0R4 release, Pulse Connect Secure is made available in AWS Market Place.

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.

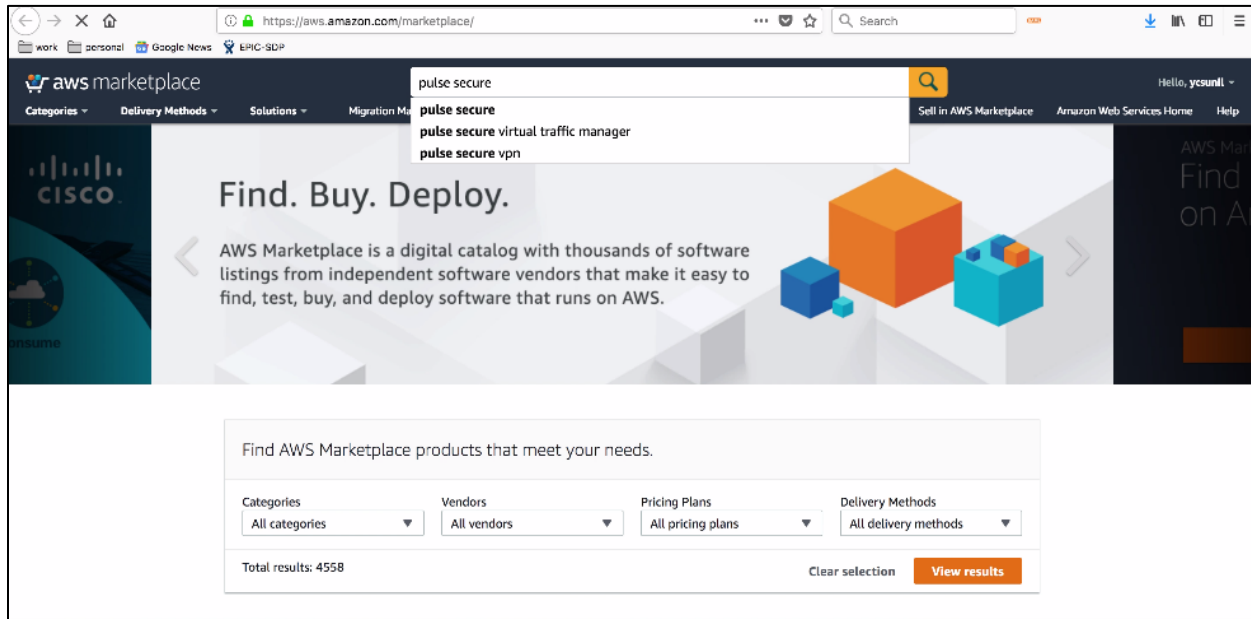


Note: From release 9.0R1 onwards, PCS supports VM with 2-NICs model and 3-NICs model for deployment.

Deploying Pulse Connect Secure on AWS Marketplace

1. Launch AWS Marketplace using the url: <https://aws.amazon.com/marketplace> and search with keyword Pulse Secure.

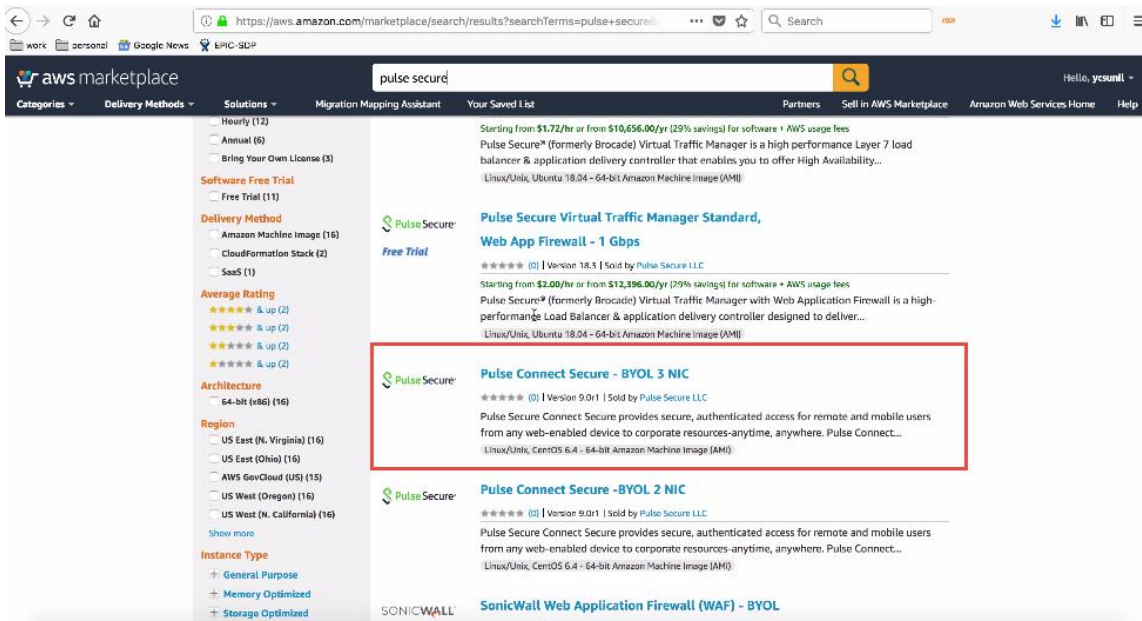
Figure 1: AWS Marketplace



AWS Marketplace contains the following two Pulse Connect Secure SKUs:

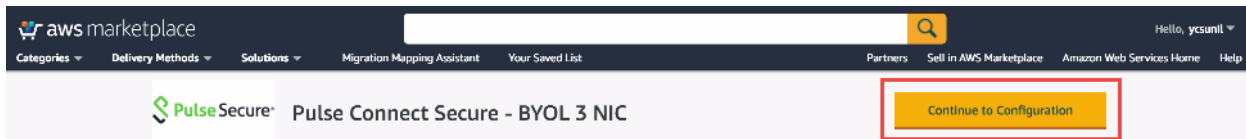
- Pulse Connect Secure - BYOL 2 NIC
- Pulse Connect Secure - BYOL 3 NIC

Figure 2: Subscribe to Pulse Connect Secure – BYOL 3 NIC



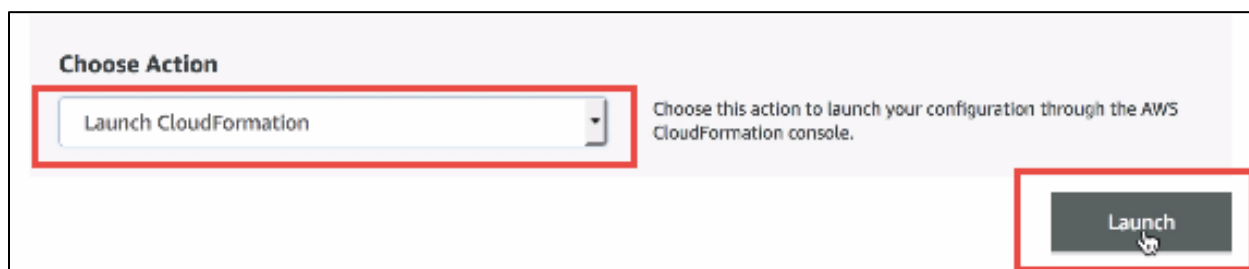
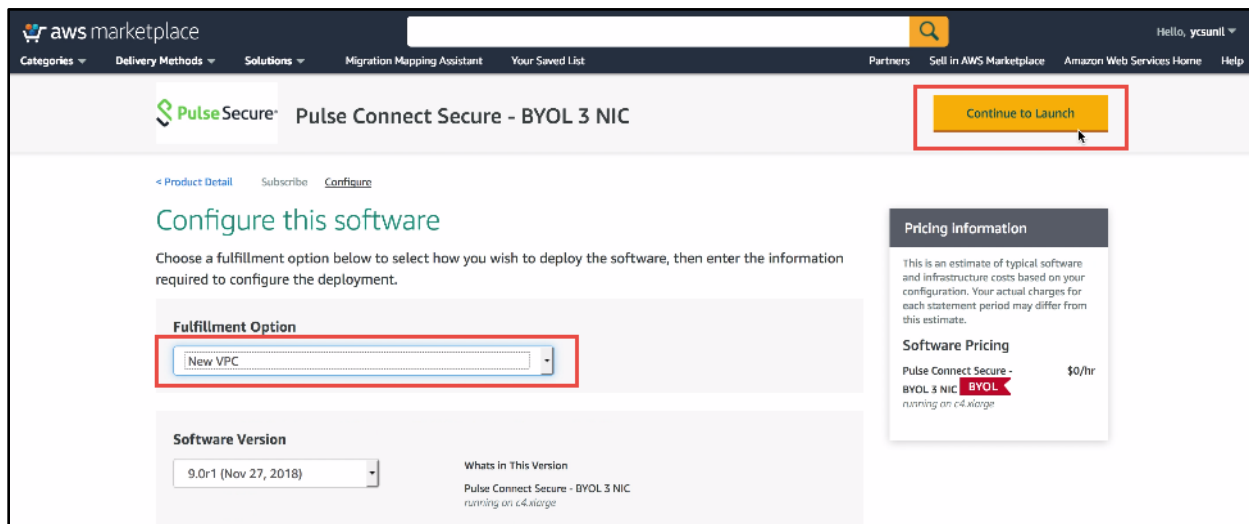


2. Select either 3-NIC model or 2-NIC model based on your requirement. In the Product Subscription page displayed, click **Continue to Subscribe**. In this section, 3-NIC model is chosen as example.
3. After subscribing, proceed to configuration by clicking **Continue to Configuration**.



4. 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**.

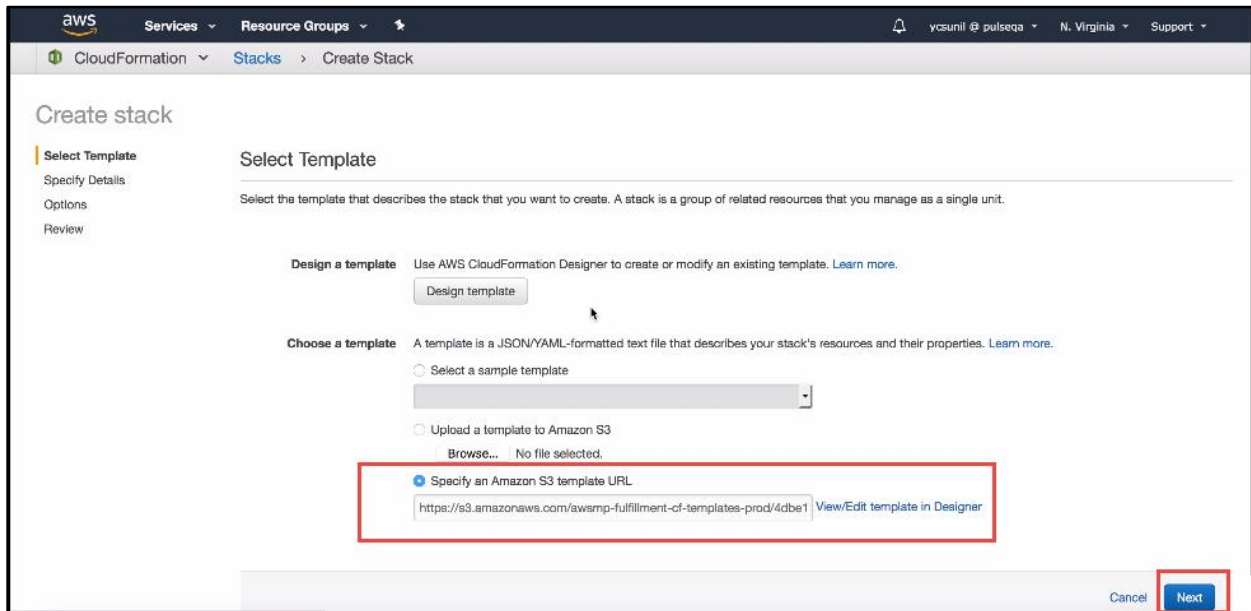
Figure 3: CloudFormation Template



Select Template

5. 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

6. In the Specify Details page, specify a name for the stack.
7. In the Parameters section, use the default parameter values. These are defined in the CloudFormation template.
8. 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](#).
 - 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>.
9. 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

Pulse Connect Secure Configuration

Software Version Pulse Connect Secure version

Instance Type Pulse Connect Secure instance type

Admin User Name Pulse Connect Secure admin user.

Admin Password Password for the Pulse Connect Secure admin user.

Config Data Pulse Connect Secure configuration data.

SSH Key Name Name of an existing EC2 KeyPair. Your PCS will launch with this KeyPair.

Security Configuration

Remote Access CIDR The CIDR IP range that is permitted to access the Pulse Connect Secure Instance

[Cancel](#) [Previous](#) [Next](#)

Review

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

ps-pcs-sa-8.3f3.0-rest-00 X Licensing X CloudFormation Main X Authcodes for VLS and X Pulse Secure Service X AWS Management Console X Create A New Stack X

work personal Google News EPIC-SDP

https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/

Security Configuration

RemoteAccessCIDR 0.0.0.0/0

[Options](#)

Tags

No tags provided

Rollback Triggers

No monitoring time provided

No rollback triggers provided

Advanced

Notification

Termination Protection Disabled

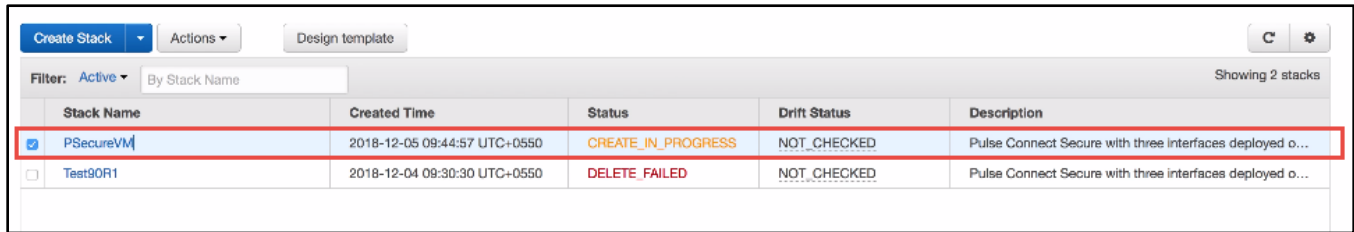
Timeout none

Rollback on failure Yes

[Quick Create Stack](#) (Create stacks similar to this one, with most details auto-populated)

[Cancel](#) [Previous](#) [Create](#)

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



	Stack Name	Created Time	Status	Drift Status	Description
<input checked="" type="checkbox"/>	PSecureVM	2018-12-05 09:44:57 UTC+0550	CREATE_IN_PROGRESS	NOT_CHECKED	Pulse Connect Secure with three interfaces deployed o...
<input type="checkbox"/>	Test90R1	2018-12-04 09:30:30 UTC+0550	DELETE_FAILED	NOT_CHECKED	Pulse Connect Secure with three interfaces deployed o...

To access Pulse Connect Secure Virtual Appliance, see [Accessing the Pulse Connect Secure Virtual Appliance](#)

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
- 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)



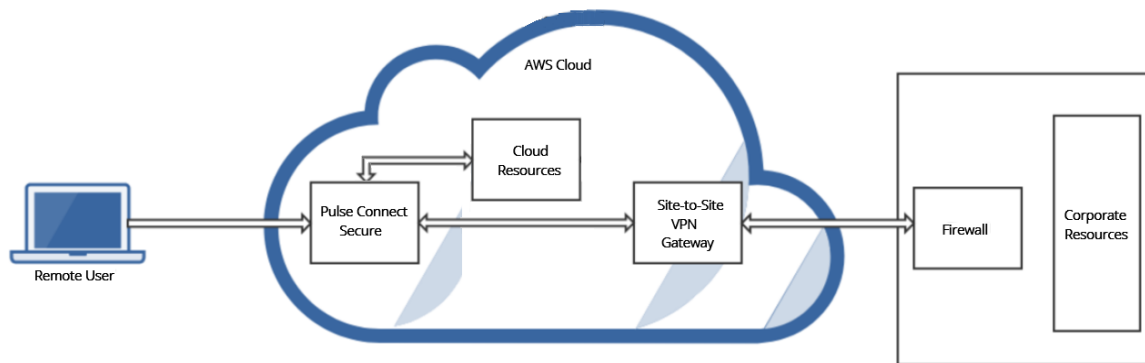
Note:

* 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

Below is the one-time activity to be followed to deploy Pulse Connect Secure on AWS.

- [Registering the AMI](#)

Below is the step to be followed for each deployment of Pulse Connect Secure.

- [Deploying Pulse Connect Secure on AWS using AWS Portal](#)

Registering the AMI

This section describes the steps to register the AMI.

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:
 - a. 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>"
  }
}
```

- b. 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.

- c. 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.

Pulse Connect Secure can be deployed:

- on [a new Virtual Private Cloud](#) or
- on [an already existing Virtual Private Cloud](#)

- as [a license server](#)
- using [vTM as load balancer](#)

Deploying PCS on New Virtual Private Cloud

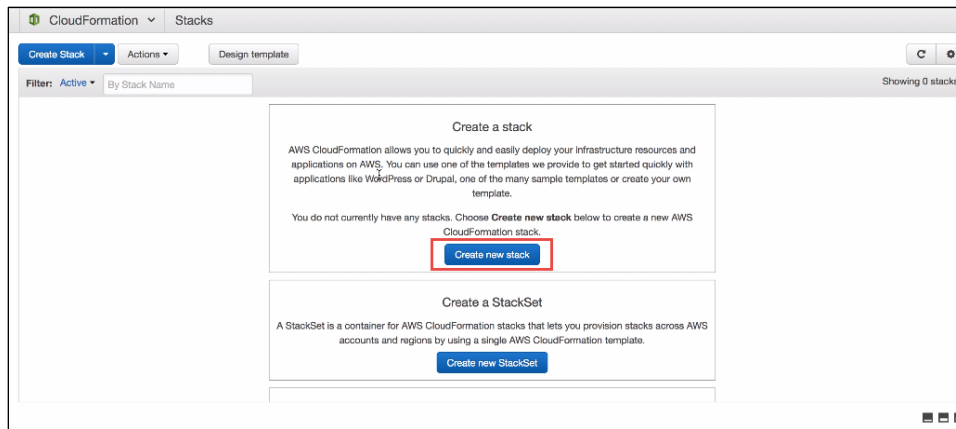
This section describes PCS deployment with [three NIC cards](#) and [two NIC cards](#).

Deployment on VM with Three NIC Cards

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

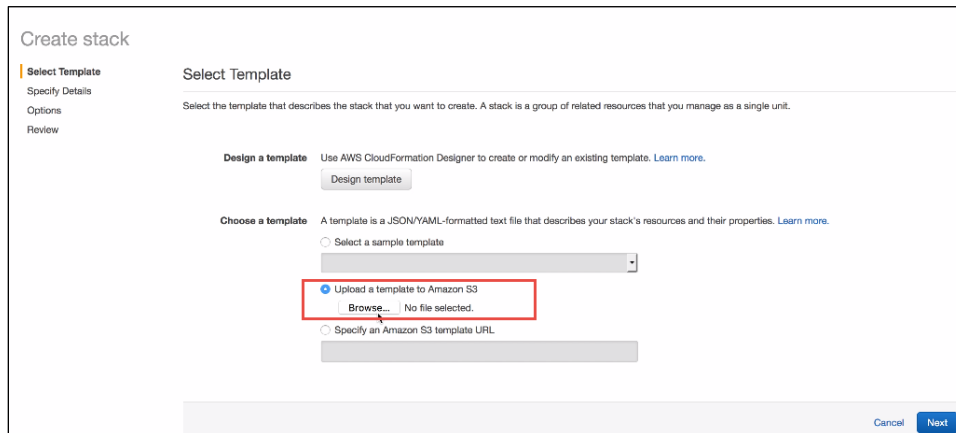
1. Select **AWS Services > CloudFormation** and click **Create new stack**.

Figure 7: Create New Stack



2. 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 8: 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 PCSCfgData is set to "y".

Figure 9: Specify Details for New Virtual Private Cloud

Specify Details
Options
Review

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name

Parameters

New VPC Configuration

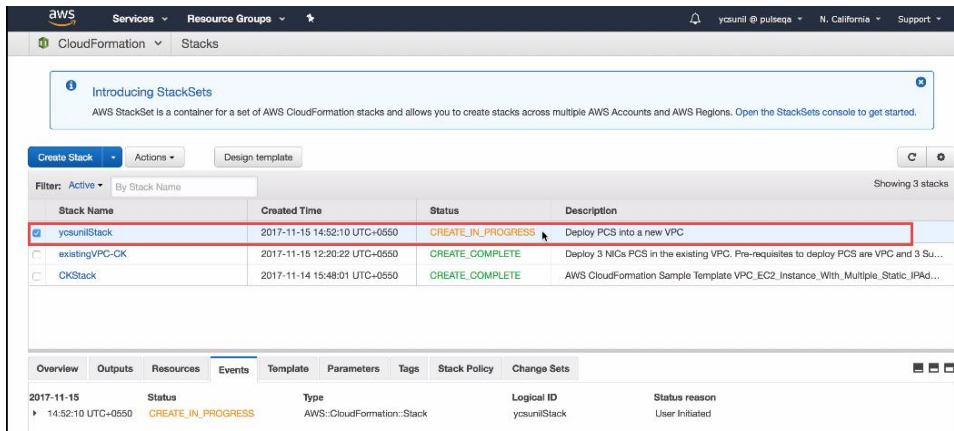
New VPC address space	<input type="text" value="10.20.0.0/16"/>	CIDR block for entire VPC.
Internal Subnet address space	<input type="text" value="10.20.1.0/24"/>	PCS internal interface connects to this subnet
External Subnet address space	<input type="text" value="10.20.2.0/24"/>	PCS external interface connects to this subnet
Management Subnet address space	<input type="text" value="10.20.3.0/24"/>	PCS management interface connects to this subnet
Tunnel Subnet address space	<input type="text" value="10.20.4.0/24"/>	For L3 VPN connections PCS hands over IP to the clients from this subnet

PCS Configuration

PCS AMI ID	<input type="text" value="ami-39407f59"/>	AMI ID of your existing PCS image
Instance Type	<input type="text" value="t2.medium"/>	Select PCS instance type
PCS Config Data	<input type="text" value="<pulse-config><primary-dns>8.8.8.8</primary-dns>"/>	PCS config data
SSH Key Name	<input type="text" value="Search"/>	Name of an existing EC2 KeyPair. Your PCS will launch with this KeyPair.

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

Figure 10: New VPC

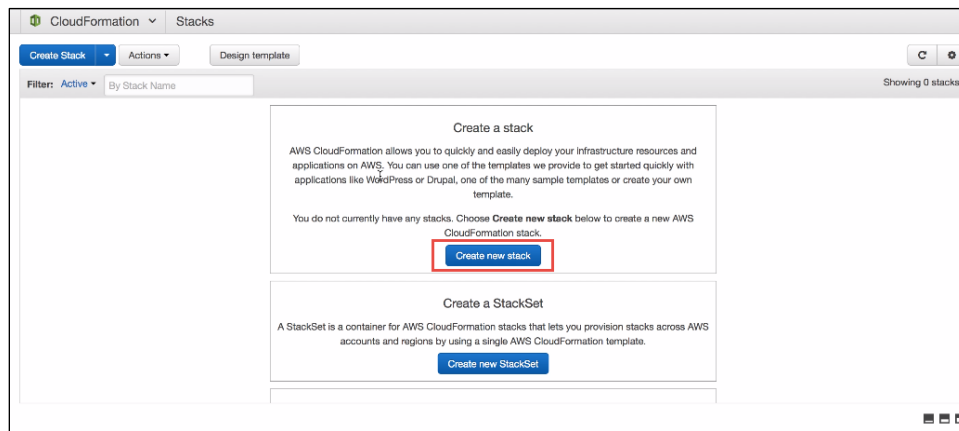


Deployment on VM with Two NIC Cards

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

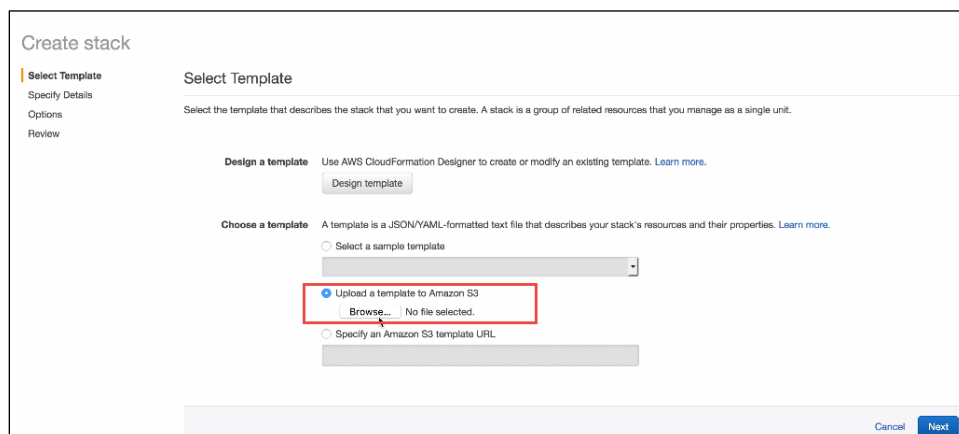
1. Select **AWS Services > CloudFormation** and click **Create new stack**.

Figure 11: 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 12: 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”.

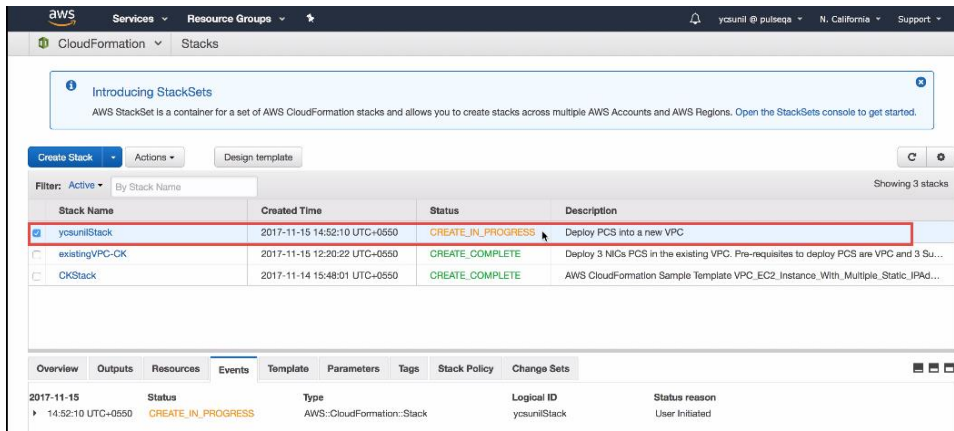
Figure 13: Specify Details for New Virtual Private Cloud

The screenshot shows the 'Specify Details' page for a new Virtual Private Cloud. The page is divided into several sections:

- Stack name:** A text input field.
- Parameters:** A section header.
- New VPC Configuration:** A section containing four sub-sections:
 - New VPC address space:** A text input field with the value '10.20.0.0/16' and a description 'CIDR block for entire VPC.'
 - Internal Subnet address space:** A text input field with the value '10.20.1.0/24' and a description 'PCS internal interface connects to this subnet.'
 - External Subnet address space:** A text input field with the value '10.20.2.0/24' and a description 'PCS external interface connects to this subnet.'
 - Tunnel Subnet address space:** A text input field with the value '10.20.4.0/24' and a description 'For L3 VPN connections PCS hands over IP to the clients from this subnet.'
- PCS Configuration:** A section containing three sub-sections:
 - PCS AMI ID:** A text input field with the value 'ami-39407f59' and a description 'AMI ID of your existing PCS image.'
 - Instance Type:** A dropdown menu with the value 't2.medium' and a description 'Select PCS instance type.'
 - PCS Config Data:** A text input field with the value '<pulse-config><primary-dns>8.8.8.8</primary-dns>' and a description 'PCS config data'. This field is highlighted with a red box.
- SSH Key Name:** A text input field with the value 'Search' and a description 'Name of an existing EC2 KeyPair. Your PCS will launch with this KeyPair.'

- **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](#).
 - **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 14: New VPC



Deploying PCS on an Existing Virtual Private Cloud

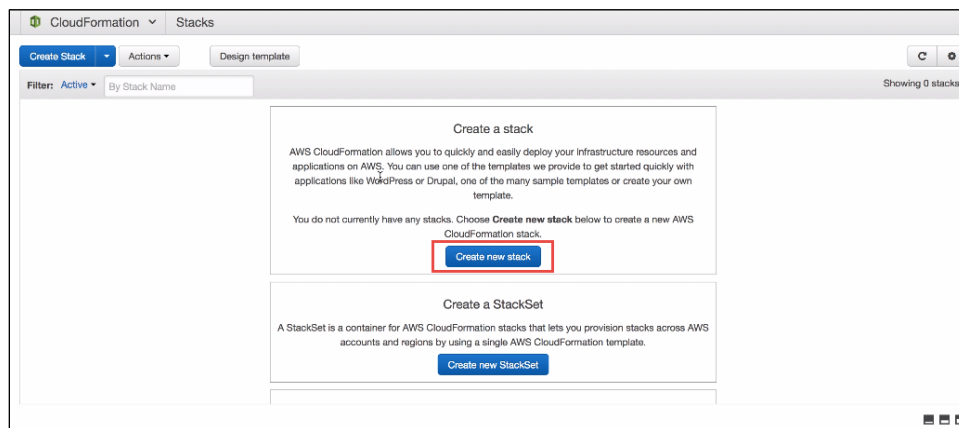
This section describes PCS deployment with [three NIC cards](#) and [two NIC cards](#).

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 15: 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 16: Upload Template

The screenshot shows the 'Create stack' wizard in AWS CloudFormation. The 'Select Template' step is active. Under 'Choose a template', the 'Upload a template to Amazon S3' radio button is selected. Below it, the 'Browse...' button is highlighted with a red rectangle. Other options include 'Select a sample template' and 'Specify an Amazon S3 template URL'. The 'Design a template' section is also visible at the top.

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 17: Specify Details for Existing Virtual Private Cloud

The screenshot shows the 'Specify Details' step of the 'Create stack' wizard. The 'Stack name' field is empty. Under the 'Parameters' section, the 'Existing VPC details' subsection contains four empty input fields: 'Existing VPC ID', 'Internal Subnet ID', 'External Subnet ID', and 'Management Subnet ID'. The 'PCS Configuration' subsection contains three fields: 'PCS AMI ID' (empty), 'Instance Type' (set to 't2.medium'), and 'PCS Config Data' (set to '<pulse-config><primary-dns>8.8.8.8/<primary>'). The 'PCS Config Data' field is highlighted with a red rectangle. The 'SSH Key Name' field is empty.

- **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

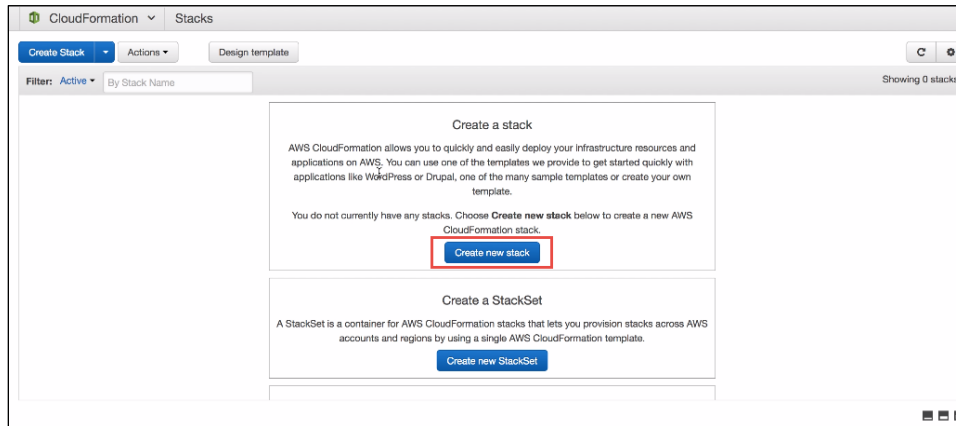
- **Management Subnet ID:** Subnet from which Pulse Connect Secure management interface needs to lease IP
 - **Tunnel Subnet ID:** Subnet which will be configured as tunnel IP pool in the Pulse Connect Secure VPN Profile
 - **PCS AMI ID:** ID of the uploaded AMI file
 - **Instance Type:** Size of the instance – t2.medium or t2.large
 - **PCS Config Data:** Provisioning parameters in an XML format. For details, see [Pulse Connect Secure Provisioning Parameters](#).
 - **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:

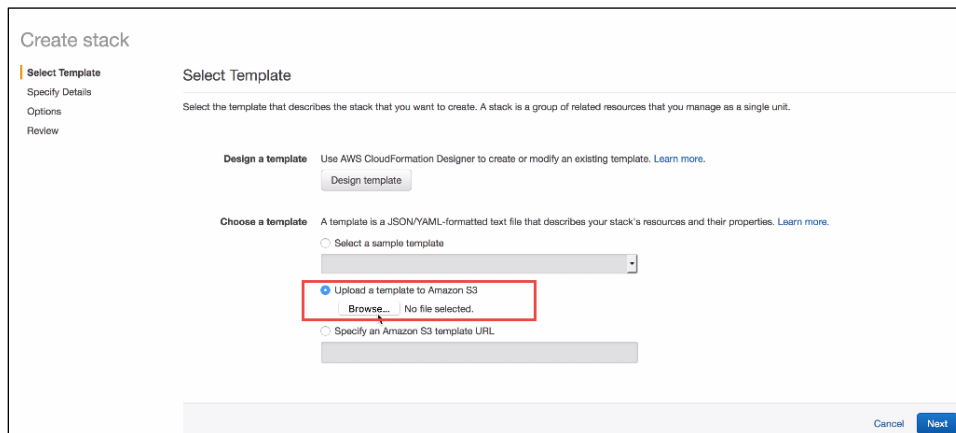
1. Select **AWS Services > CloudFormation** and click **Create new stack**.

Figure 18: 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 19: 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 20: Specify Details for Existing Virtual Private Cloud

The screenshot shows the 'Specify Details' page in the AWS CloudFormation console. The page is titled 'Create stack' and has a sidebar with 'Specify Details' selected. The main content area is divided into sections: 'Specify Details' (with a 'Stack name' field), 'Parameters' (with 'Existing VPC details' and 'PCS Configuration' subsections), and 'Existing VPC details' (with 'Existing VPC ID', 'Internal Subnet ID', and 'External Subnet ID' fields). The 'PCS Configuration' section includes 'PCS AMI ID', 'Instance Type' (set to 't2.medium'), 'PCS Config Data' (set to '<pulse-config><primary-dns>8.8.8.8/</primary>'), and 'SSH Key Name' (with a search dropdown). The 'PCS Config Data' field is highlighted with a red box.

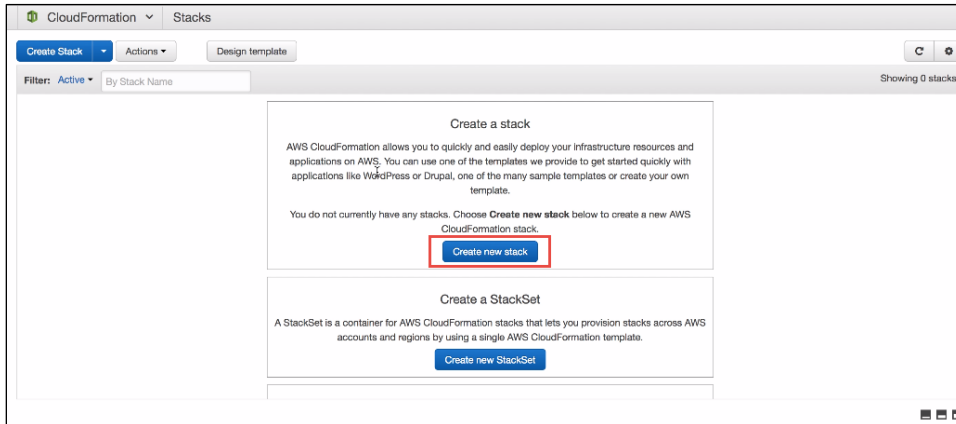
- **Stack name:** Specify the stack name in which Pulse Connect Secure needs to be deployed
 - **Existing VPC ID:** Virtual private cloud ID
 - **Internal Subnet ID:** Subnet from which Pulse Connect Secure internal interface needs to lease IP
 - **External Subnet ID:** Subnet from which Pulse Connect 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:

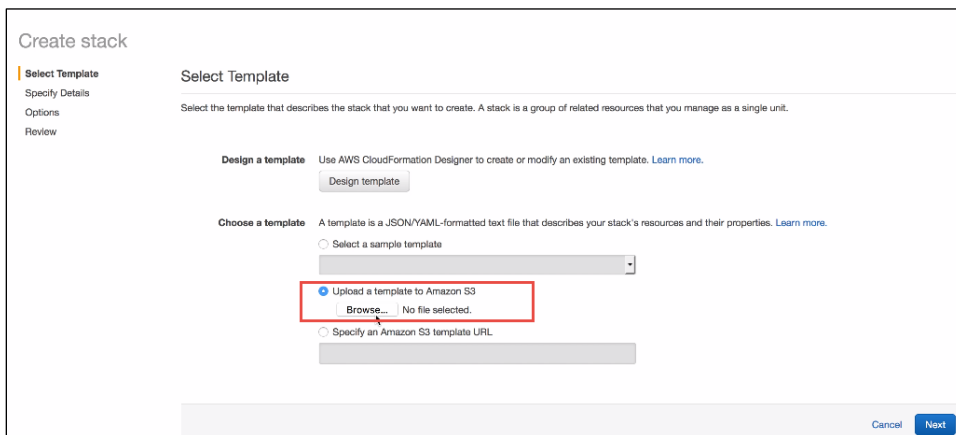
1. Select **AWS Services > CloudFormation** and click **Create new stack**.

Figure 21: 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 22: Upload Template



- 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 23: Specify Details for Existing Virtual Private Cloud

CloudFormation > Stacks > Create Stack

Create stack

Select Template
Specify Details
Options
Review

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name

Parameters

Existing VPC details

Existing VPC ID ID of existing VPC

Internal Subnet ID ID of the subnet where PCS internal interface connects

External Subnet ID ID of the subnet where PCS External interface connects

Management Subnet ID ID of the subnet where PCS Management interface connects

PCS Configuration

PCS AMI ID AMI ID of your existing PCS image

Instance Type Select PCS instance type

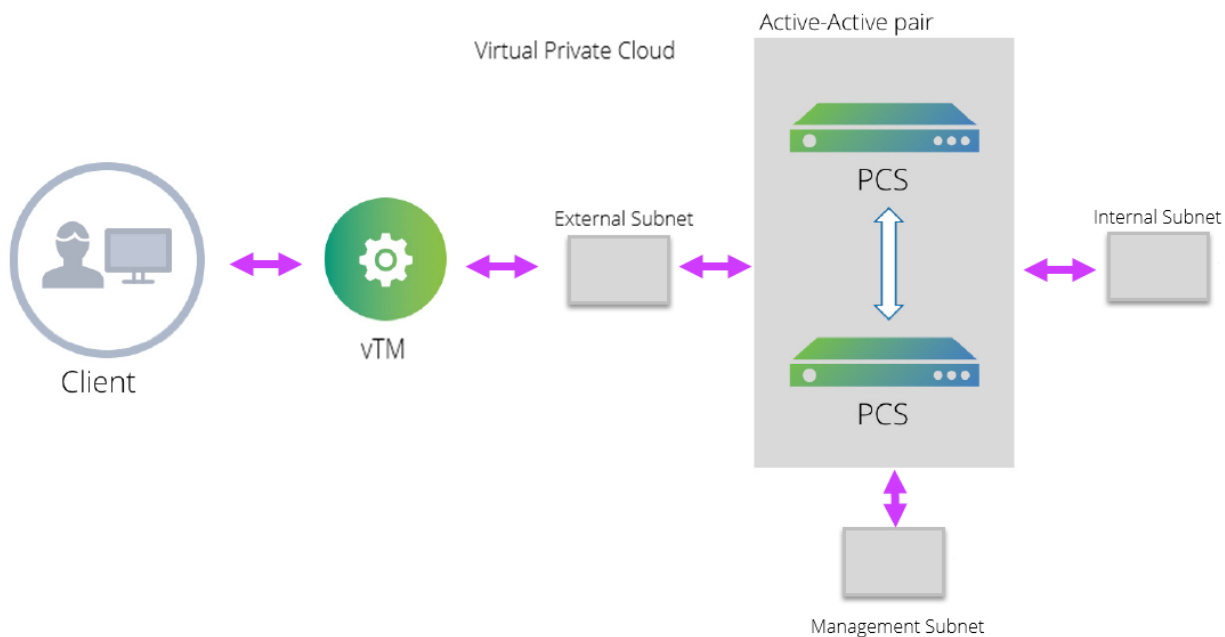
PCS Config Data PCS config data

SSH Key Name Name of an existing EC2 KeyPair. Your PCS will launch with this KeyPair.

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 24: Deploying PCS A-A Cluster Topology Diagram



The deployment process involves the following steps:

- [Deploying Two PCS EC2 instances Using CloudFormation Template](#)
- [Forming the Active-Active Cluster](#)
- [Deploying Virtual Traffic Manager EC2 Instance in the External Subnet of PCS in AWS](#)
- [Setting Up and Configuring vTM for External Users](#)

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 25: PCS A-A Cluster Status

The screenshot shows the Pulse Secure web interface. The top navigation bar includes links for System, Authentication, Administrators, Users, Maintenance, and Wizards. The main content area is titled 'Clustering > Cluster Status'. Under 'Cluster Status', there are tabs for 'Status' and 'Properties'. The 'Status' tab is active, displaying the following information:

- Cluster Name: AWS_AA
- Type: VA-SPE
- Configuration: Active/Active

Below this information are buttons for 'Add Members...', 'Enable', 'Disable', and 'Remove'. A dropdown menu shows '10 records per page'. A search bar is located on the right. The main table lists the cluster members:

	Member Name	Internal Address	External Address	Status	Notes	Sync Rank	Update
<input type="checkbox"/>	PCS1	10.251.1.214/24	10.251.2.180/24	●	Enabled	0	
<input type="checkbox"/>	PCS2	10.251.1.238/24	10.251.2.143/24	●	Leader	0	

At the bottom right, there are navigation links: '← Previous', '1', and 'Next →'.

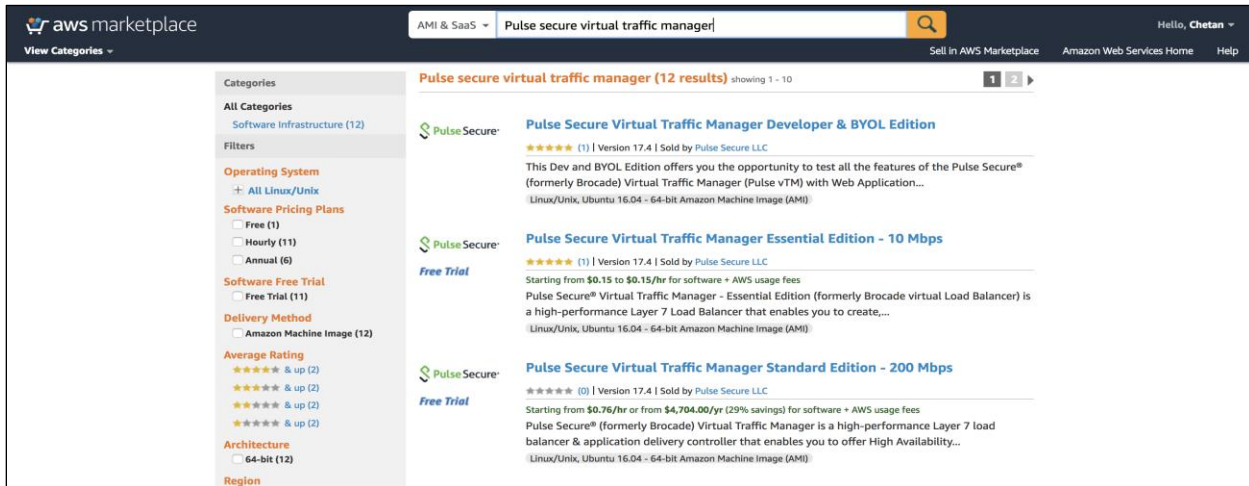
Deploying Virtual Traffic Manager EC2 Instance in the External Subnet of PCS in AWS

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

To deploy through Marketplace, follow the below steps:

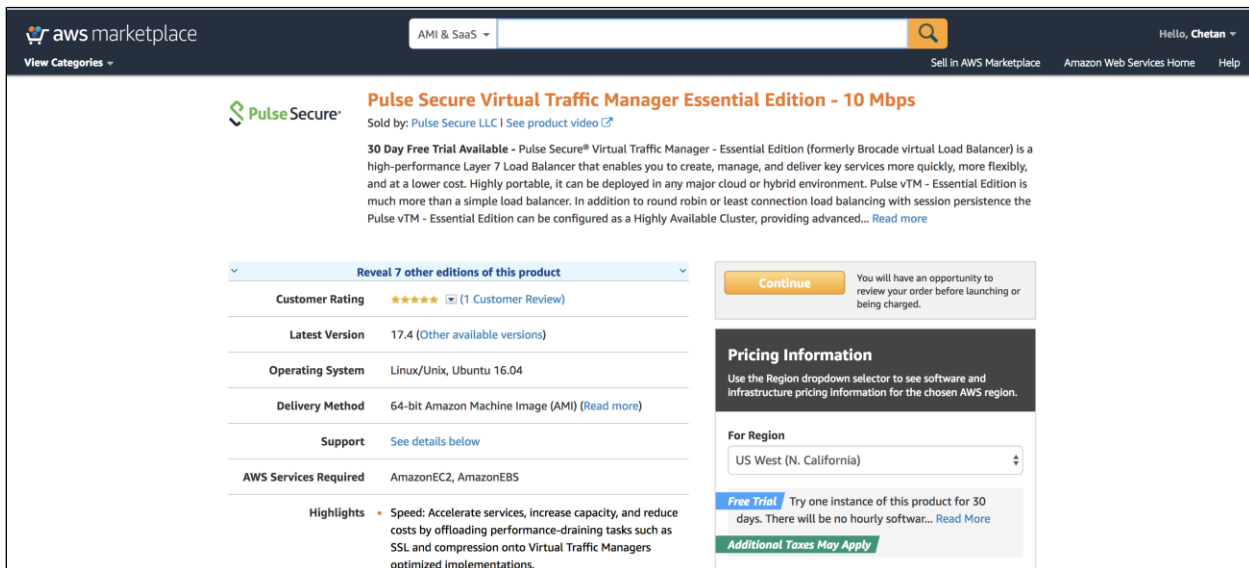
1. Search and select **Pulse Secure vTM** in AWS Marketplace.

Figure 26: AWS Marketplace > Pulse Secure vTM



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

Figure 27: 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 28: 1-Click Launch Tab

Launch on EC2:
Pulse Secure Virtual Traffic Manager Essential Edition - 10 Mbps

1-Click Launch (Review, modify and launch) | **Manual Launch** (With EC2 Console, API or CLI) | **Service Catalog** (Copy to SC and Launch)

Click "Launch with 1-Click" to launch this software with the settings below
The default settings are provided by the software seller and AWS Marketplace.

- Version:** 17.4, released 11/07/2017
- Region:** US West (N. California)
- EC2 Instance Type:** r3.xlarge
- VPC Settings:** Will launch into EC2 Classic
- Security Group:** Create new based on seller settings
- Key Pair:** aijoseph

Price for your Selections:

- \$3.11 / hour
\$2.96 r3.xlarge EC2 Instance usage fees + \$0.15 hourly software fee
Additional taxes may apply.
- \$0.08 per GB-month of provisioned storage
EBS Magnetic volumes
- \$0.08 per 1 million I/O requests
EBS Magnetic volumes

Launch with 1-click

You will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's [End User License Agreement \(EULA\)](#) and your use of AWS services is subject to the [AWS Customer Agreement](#).

Cost Estimator

- \$2,242.08 / month
Additional taxes may apply.
r3.xlarge EC2 Instance usage fees
Assumes 24 hour use over 30 days

Software Charges

- \$108.00 / month
\$108.00 monthly software fees for r3.xlarge

To deploy vTM through 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 29: Pulse Secure vTM Login Page

PulseSecure® Virtual Traffic Manager Appliance 500 L 10 17.4

Login

Pulse Secure vTM Administration Server

Software: **Virtual Traffic Manager Appliance 17.4**

Use of this software is subject to the Pulse Secure Terms and Conditions of Sale.
Please review these terms, published at **Pulse Secure Terms and Conditions of Sale** before proceeding.

Login to 10.251.2.152

Enter a username and password to access the administration server.

Your session timed out. Please login.

Username:

Password:

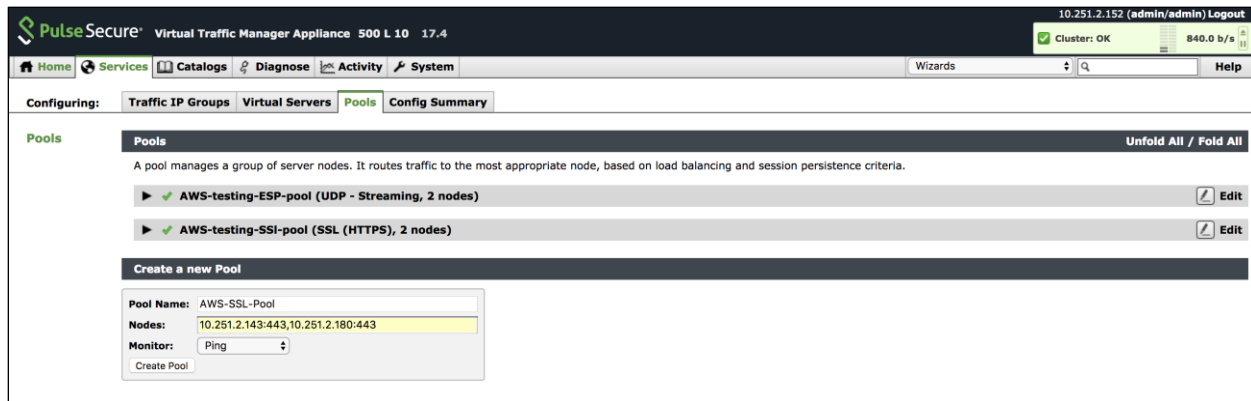
Next step is to set up the vTM for the external users using traffic pools and load balancing virtual servers. Traffic pool is the group that will bind to virtual server for load balancing. In an Active-Active Cluster scenario, traffic pool comprises cluster nodes. We need to create two separate traffic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

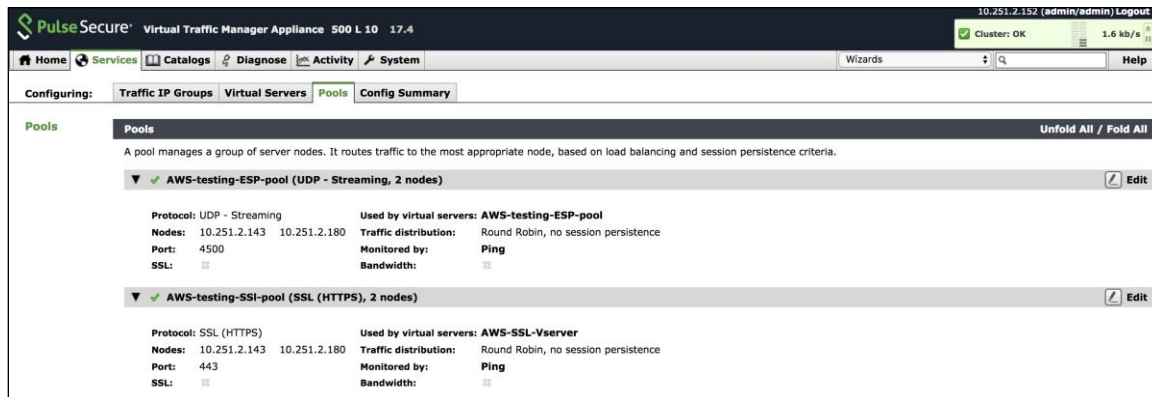
Complete these steps for SSL and UDP. For details, refer to the section “Creating PCS Pools” in [Load Balancing PCS with vTM Deployment Guide](#).

Figure 30: Create Traffic Pool



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.

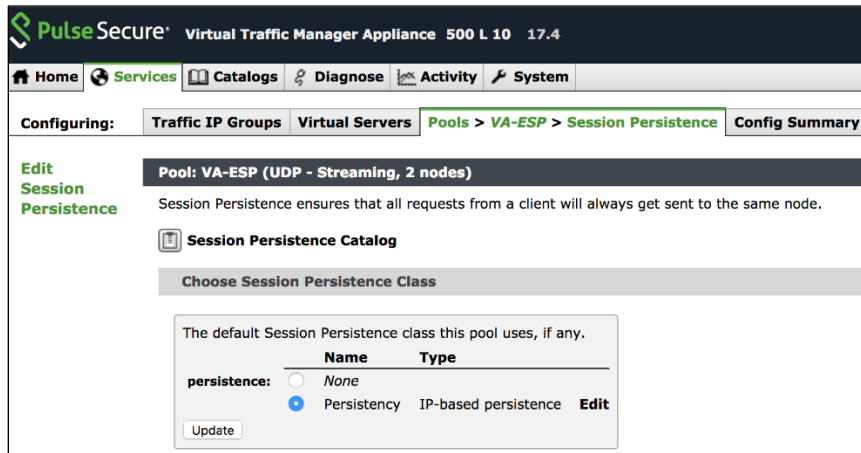
Figure 31: SSL and UDP Pools



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 32: 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 33: Create Virtual Server

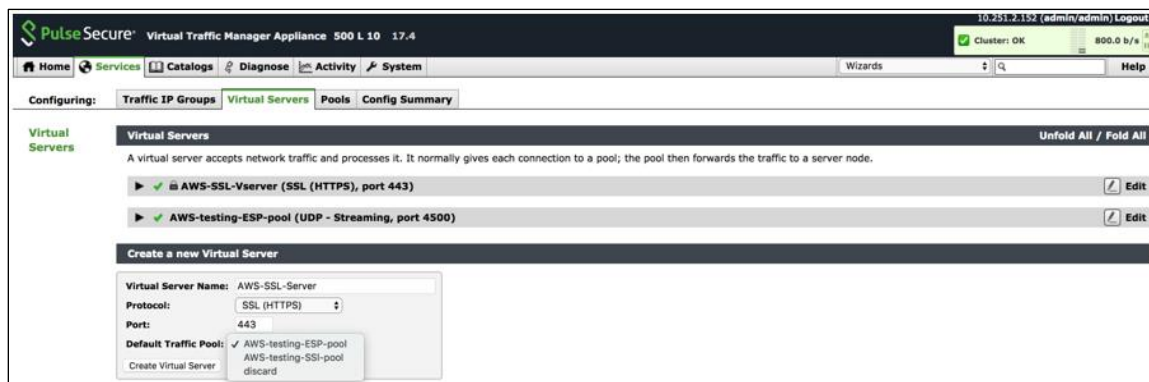
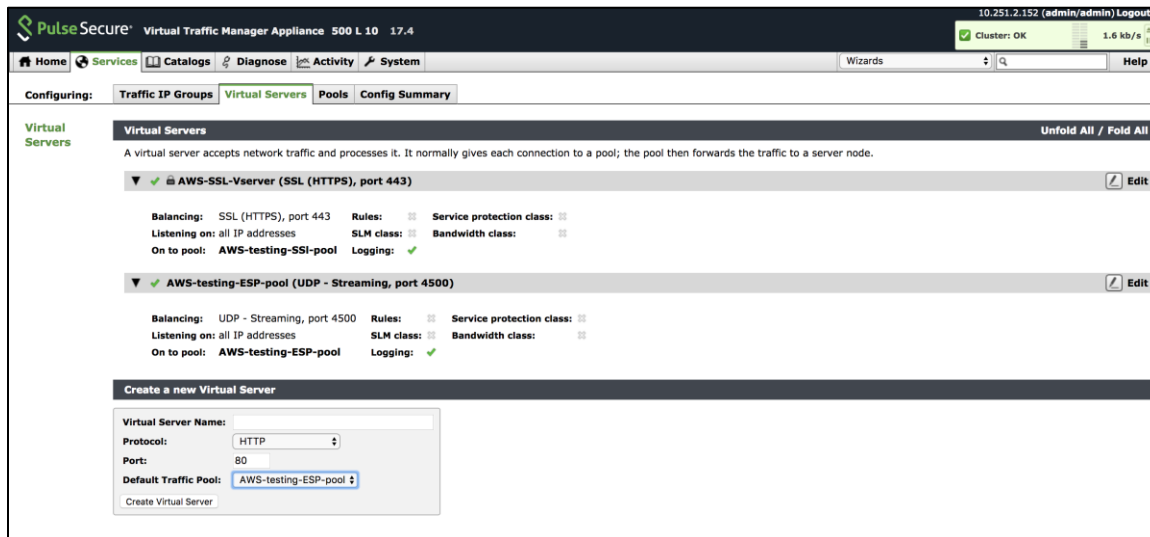
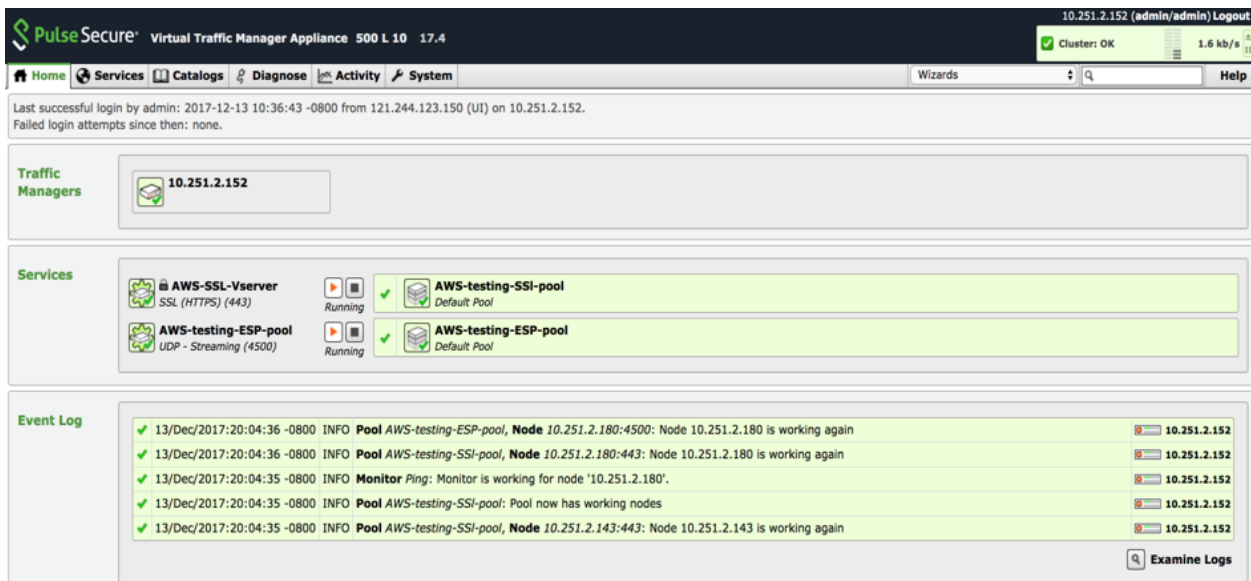


Figure 34: Virtual Servers to Handle SSL and UDP Traffic



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

Figure 35: 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 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 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.

Provisioning Pulse Connect Secure with Predefined Configuration

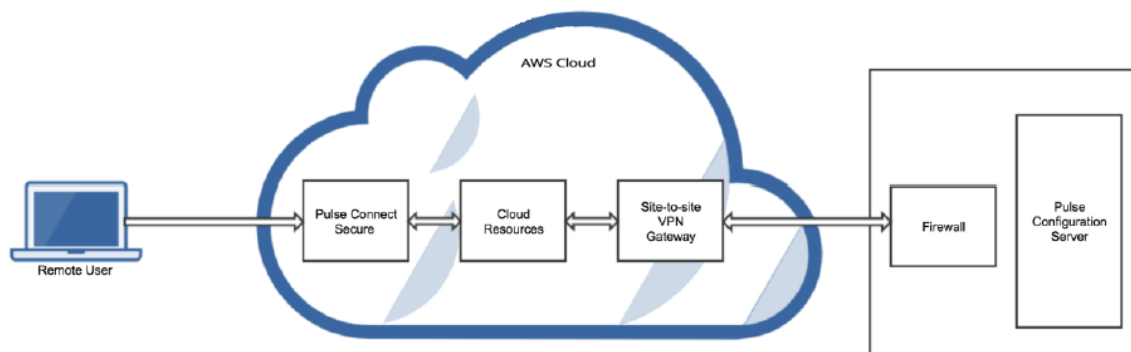
The Pulse Connect Secure Virtual Appliance can be provisioned on AWS with a predefined Pulse Connect Secure configuration. The provisioning can be done in the following two ways:

- Pulse Connect Secure administrator needs to provide the location of the XML-based configuration as a provisioning parameter. Refer '[Pulse Connect Secure Provisioning Parameters](#)' 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 36: Pulse Configuration Server in Corporate Network



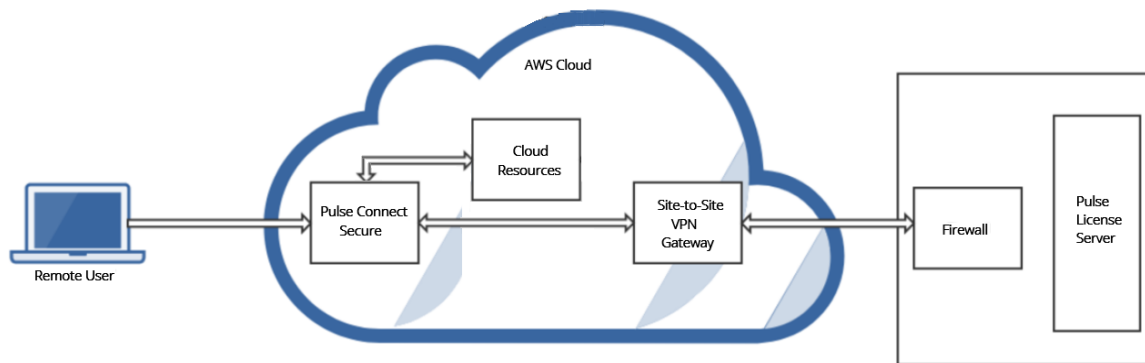
Configuring Licenses on the Pulse Connect Secure Appliance

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

The Pulse License server can be made available in the [corporate network](#)

Pulse License Server in Corporate Network

Figure 37: Pulse License Server in a Corporate Network



Pulse License Server in Cloud Network

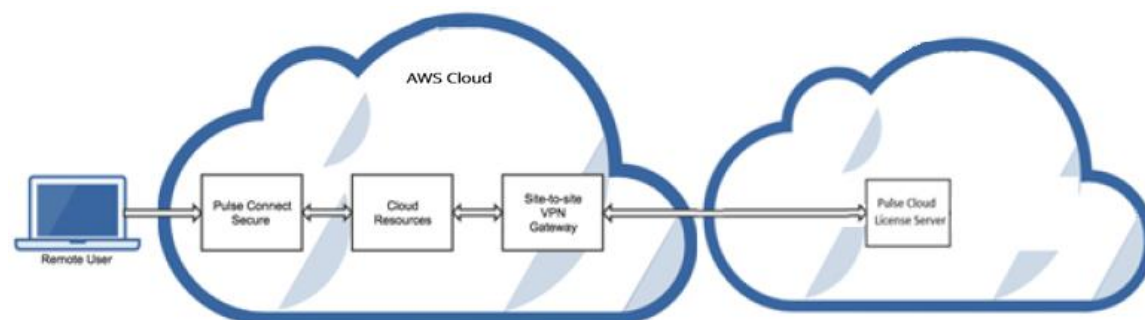
In 8.3R3, the Pulse Connect Secure virtual machines (VM) are enabled to provision licenses through the Pulse Cloud Licensing Service (PCLS). For this, administrator needs to obtain an Authentication code from Pulse Secure Support and apply it in Download Licenses page of PCS admin console. The PCS also 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>val.psecure.net</cert-common-name><cert-random-text>fdsfjpsionvsfnms</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](#)
- [Including Authentication Code in CloudFormation Template](#)

Figure 38: Pulse License Server in Cloud Network

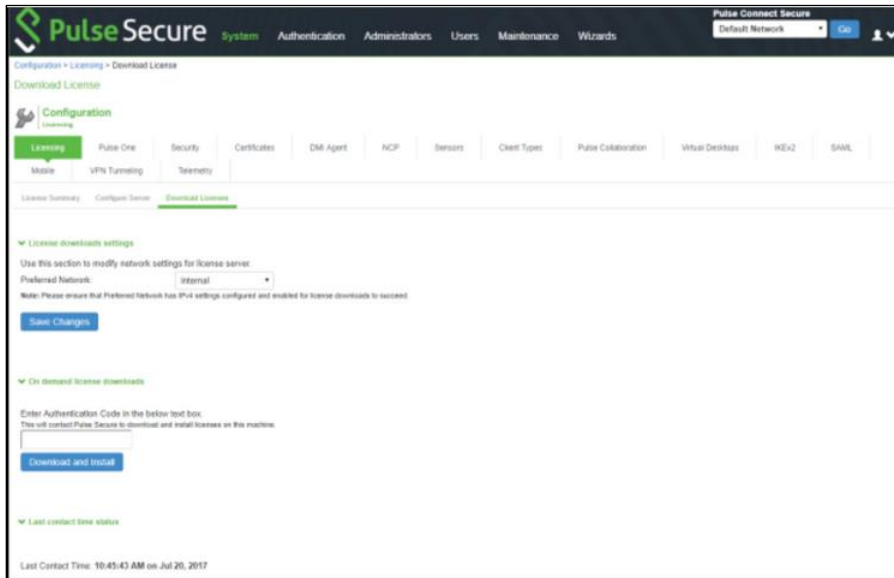


Adding Authentication Code in PCS Admin Console

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 39: Enter Authentication Code



Including Authentication Code in CloudFormation Template

To include Authentication code in the CloudFormation template:

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

For details about the license configuration, refer to [License Configuration Guide](#).

Accessing the Pulse Connect Secure Virtual Appliance

The Pulse Connect Secure virtual appliance can be accessed:

- [as an administrator](#)
- [as an end user](#)
- [using SSH console](#)

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 40.

Figure 40: 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 External Interface details	

Use the credentials provided in the provisioning parameters to log in as the administrator. The default PCS admin UI user configured in the CloudFormation 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, AWS network administrator needs to ensure that all other resources have Pulse Connect Secure Internal interface as its default gateway.

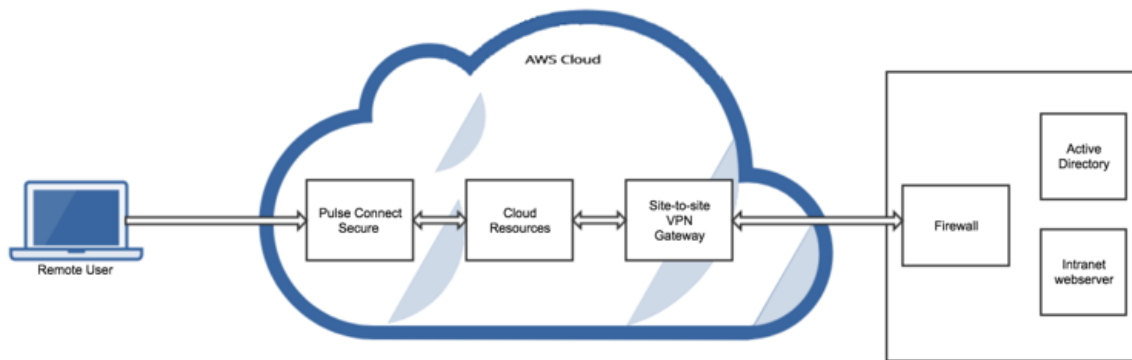
Accessing the Pulse Connect Secure Virtual Appliance as an End User

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

Figure 41: 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 External Interface details	

Figure 42: 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

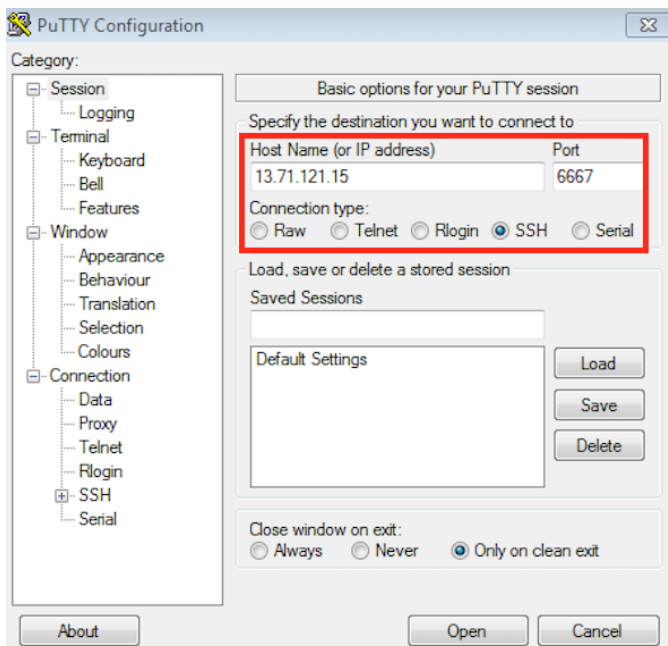
Execute the following command:

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

On Windows

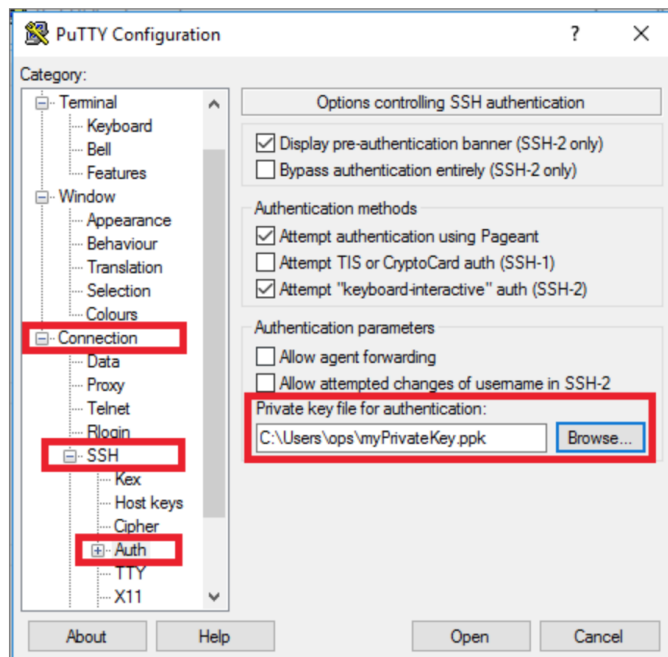
1. Launch the Putty terminal emulator.
2. In the Session category:
 - Enter the host name or IP address.
 - Enter the port number.
 - Select the connection type as SSH.

Figure 43: PuTTY Configuration – Basic Options



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

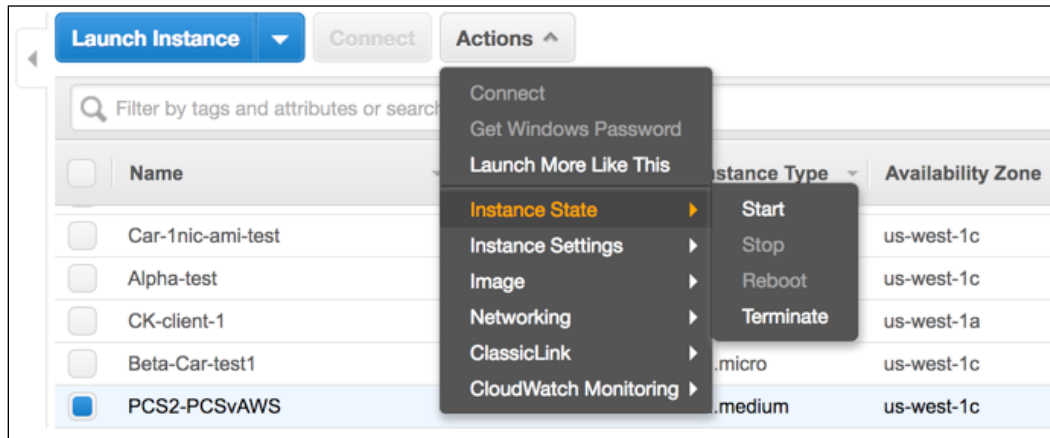
Figure 44: PuTTY Configuration – SSH Authentication



System Operations

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

Figure 45: 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": "PCSCfgData"} ] ] }}
  }
},
```

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:

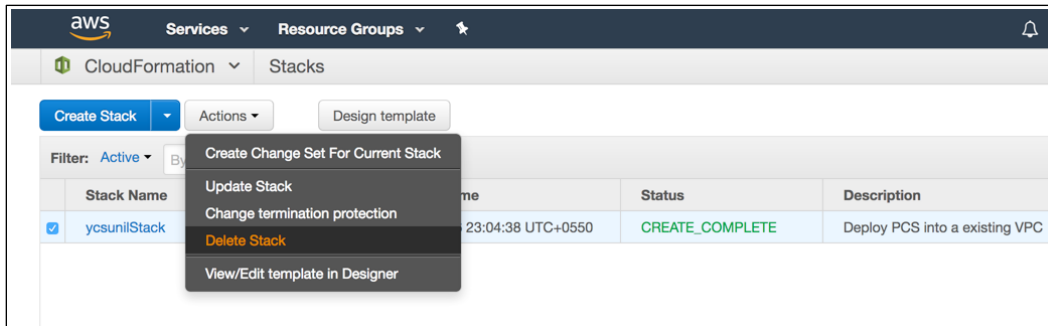
Interface Name	PCS Interface	Network ID
eth0	internal interface	nic1
eth1	external interface	nic2
eth2	management interface	nic3

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 46: 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, PCSEternalNIC and PCSManagementNIC	Networking	No
Three Elastic 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

Frequently Asked Questions

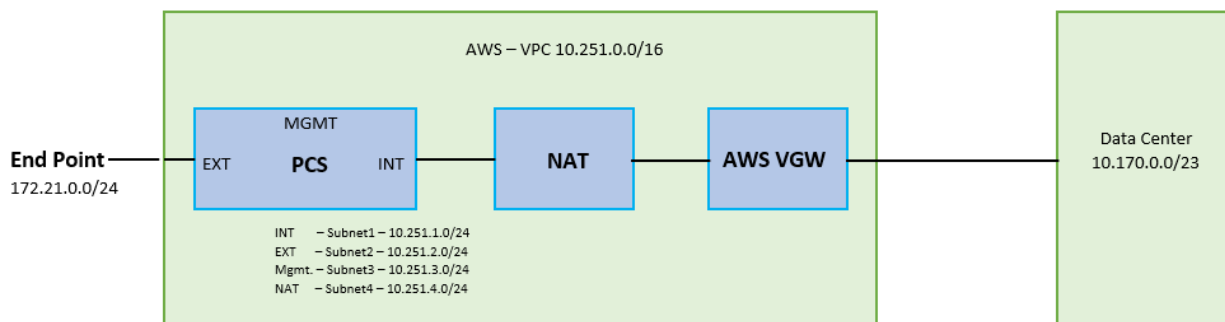
FAQ1: Packets transmitted from PCS Internal Interface are getting dropped by AWS Virtual Gateway in L3 traffic.

Cause: The packets are dropped because the source IP and MAC address are not matching and the transit routing is not supported.

Solution:

The following topology shows PCS Virtual Network in AWS Cloud connected to NAT device which in turn is connected to AWS Virtual Gateway. The AWS Virtual Gateway has the connectivity to Data Center. Here, the packets received from the PCS Internal interface are source NATed and then sent to the AWS Virtual Gateway.

Figure 49: Topology diagram



The AWS VPC has four subnets – subnet1 to subnet3 connected to PCS's Internal, External, and Management interfaces respectively, and subnet4 connected to NAT device.

Figure 50: Subnets

The screenshot shows the AWS VPC Dashboard for VPC vpc-f5142691. The subnets are listed in the following table:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability
ycsunil-subnet-3	subnet-a754b8fc	available	vpc-f5142691 ycsunil	10.251.3.0/24	250		us-west-1c
ycsunil-subnet-4	subnet-af17eaf4	available	vpc-f5142691 ycsunil	10.251.4.0/24	250		us-west-1c
ycsunil-subnet-1	subnet-5a56ba01	available	vpc-f5142691 ycsunil	10.251.1.0/24	245		us-west-1c
ycsunil-subnet-2	subnet-ae54b8f5	available	vpc-f5142691 ycsunil	10.251.2.0/24	247		us-west-1c

Route Tables in AWS for Source NATing

In the VPC, two route tables are created. The first route table is associated with the three subnets, subnet1 to subnet 3, that are connected to the PCS's three interfaces.

Figure 51: Route Table

VPC Dashboard

Filter by VPC: vpc-f5142691

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Create Route Table Delete Route Table Set As Main Table

Search Route Tables and their X

Name	Route Table ID	Explicitly Associat	Main	VPC
YC_RT	rtb-5e8e4039	1 Subnet	No	vpc-f5142691 ycsunil
ycsunil-route-table	rtb-0f34e768	3 Subnets	Yes	vpc-f5142691 ycsunil

rtb-0f34e768 | ycsunil-route-table

Summary Routes Subnet Associations Route Propagation Tags

Edit

Subnet	IPv4 CIDR	IPv6 CIDR
subnet-5a56ba01 ycsunil-subnet-1	10.251.1.0/24	-
subnet-ae54b8f5 ycsunil-subnet-2	10.251.2.0/24	-
subnet-a754b8fc ycsunil-subnet-3	10.251.3.0/24	-

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

This route table has three routes. The Rules for these routes define to which destination the packets should be routed.

10.251.0.0/18	This is the VPC local route which is created by default when the VPC is created.
0.0.0.0	If the packet is destined to the internet, then the packet is routed to internet gateway.
10.170.0.0/23	If the packet is destined to the on-premise network, then the packet is routed to the NAT device.

Figure 52: Routes

rtb-0f34e768 | ycsunil-route-table

Summary Routes Subnet Associations Route Propagation Tags

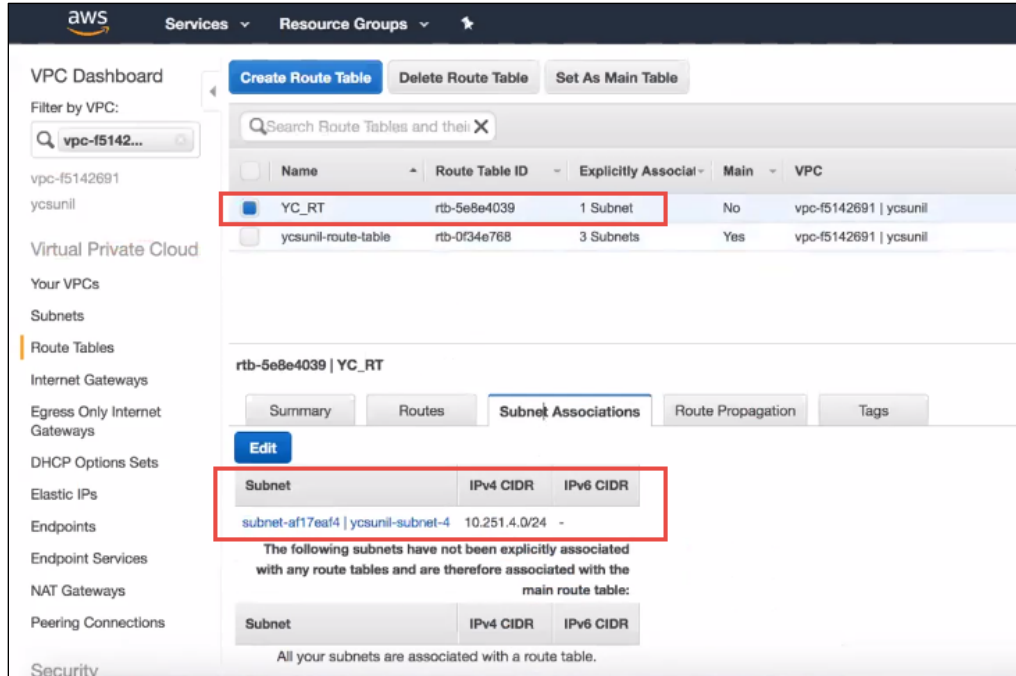
Edit

View: All rules

Destination	Target	Status	Propagated
10.251.0.0/18	local	Active	No
0.0.0.0/0	igw-203fe244	Active	No
10.170.0.0/23	eni-190dec1c / i-Oa7c9c36ea4922350	Active	No

The second route table is associated with subnet4 that is connected to the NAT device.

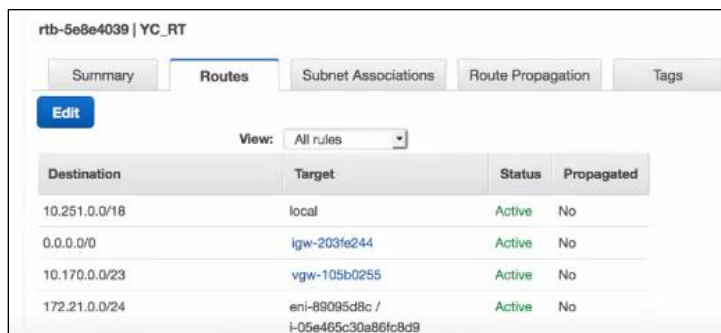
Figure 53: Route Table



This route table has four routes.

10.251.0.0/18	This is the VPC local route which is created by default when the VPC is created.
0.0.0.0	If the packet is destined to the internet, then the packet is routed to internet gateway.
10.170.0.0/23	If the packet is destined to the on-premise network, then the packet is routed to AWS Virtual Gateway.
172.21.0.0/24	The response packet received from the Data Center will have tunnel IP of End Point as the destination IP. So, if the packet is destined to the tunnel IP port, the packet is routed to the PCS Internal interface.

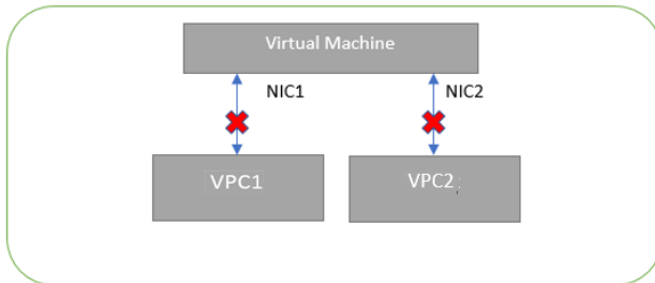
Figure 54: Routes



Appendix A: Security Group (SG)

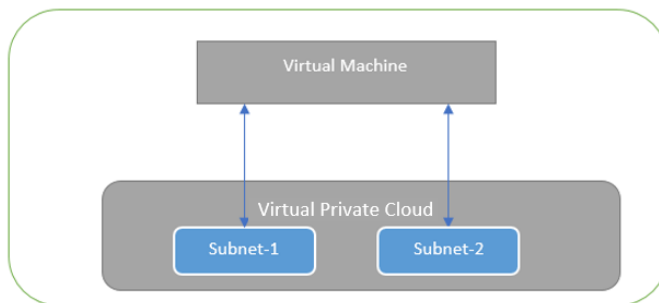
AWS has a limitation where virtual machine with multiple network interfaces cannot connect to different Virtual Private Cloud (VPCs). For example, a VM with two NICs, NIC1 and NIC2, will not be able to connect to VPC1 and VPC2 respectively.

Figure 55: Virtual Machine with two NICs Connecting to VPC1 and VPC2



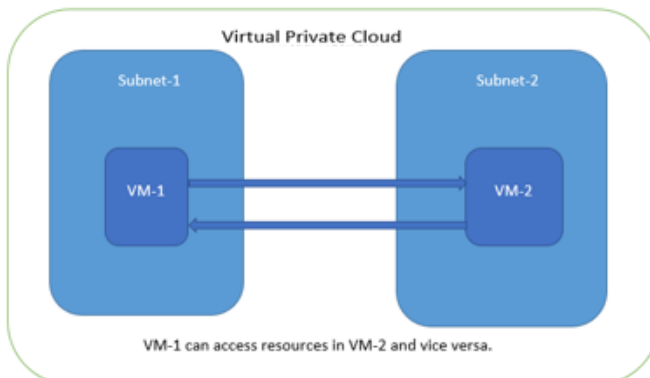
AWS supports a virtual machine with multiple NICs to connect to different Subnets under a same Virtual Private Cloud. For example, a VM with two NICs, NIC1 and NIC2, can connect to 'Subnet1' and 'Subnet2' where these subnets exist under a same Virtual Private Cloud respectively.

Figure 56: Virtual Machine with two NICs Connecting to Subnet1 and Subnet2



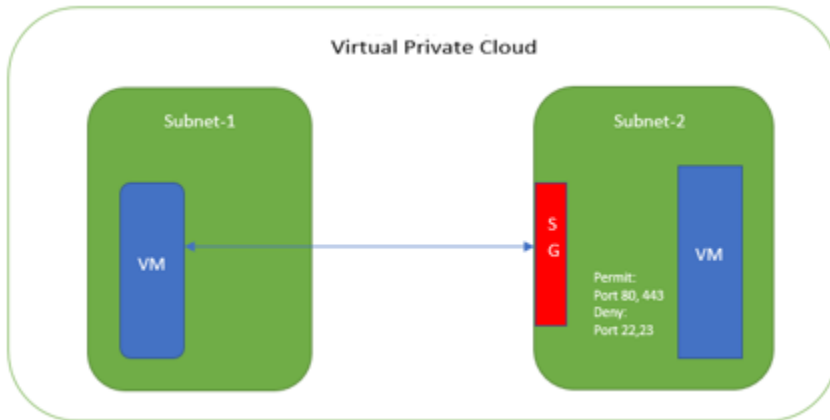
AWS provides isolation between different VPCs. But it does not provide the same kind of isolation when it comes to subnets in the same VPC. For example, consider a VPC has two subnets, Subnet1 and Subnet2. And consider two VMs, VM-1 and VM-2, which are connected to Subnet1 and Subnet2 respectively. In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

Figure 57: Virtual Machine VM-1 can Access Resources in VM-2 and Vice Versa



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

Figure 58: Traffic Filtering by AWS Support Group



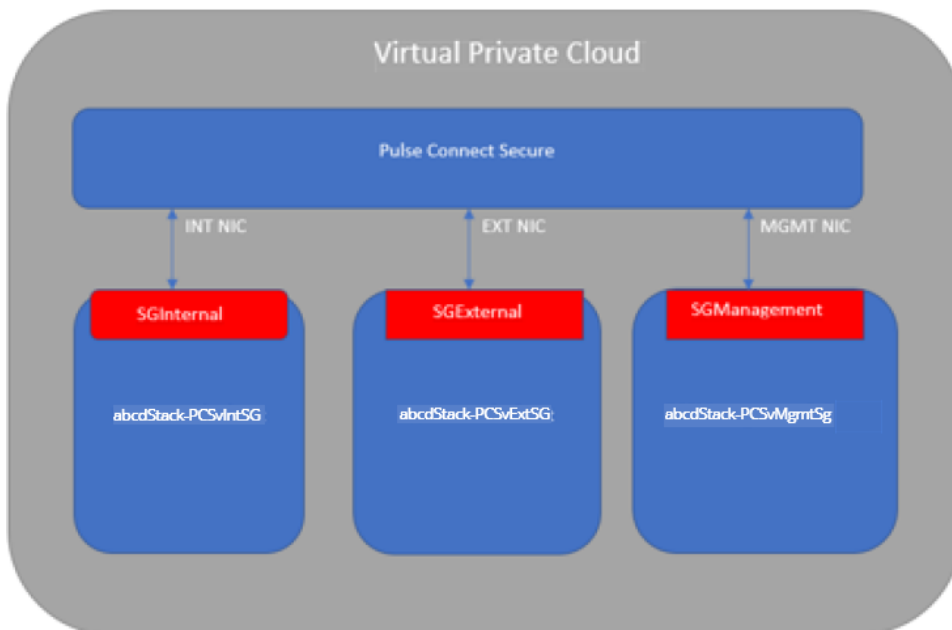
Pulse Connect Secure, when provisioned through the CloudFormation template provided by Pulse Secure, creates four subnets under a virtual private cloud named “PCSVirtualNetwork”. The four Subnets are:

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

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

1. SGExternalSubnet
2. SGInternalSubnet
3. SGManagementSubnet

Figure 59: 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 60: Stack-PCSVExtSG - Inbound Rules

sg-49208230 | sgssgilStack-PCSVExtSG

Summary Inbound Rules Outbound Rules Tags

Edit

Type	Protocol	Port Range	Source	Description
HTTP (80)	TCP (6)	80	0.0.0.0/0	
Custom TCP Rule	TCP (6)	11000-11099	0.0.0.0/0	
Custom TCP Rule	TCP (6)	6667	0.0.0.0/0	
Custom UDP Rule	UDP (17)	4500	0.0.0.0/0	
HTTPS (443)	TCP (6)	443	0.0.0.0/0	
All ICMP - IPv4	ICMP (1)	ALL	0.0.0.0/0	

Figure 61: Stack-PCSVExtSG - Outbound Rules

sg-49208230 | sgssgilStack-PCSVExtSG

Summary Inbound Rules Outbound Rules Tags

Edit

Type	Protocol	Port Range	Destination	Description
ALL Traffic	ALL	ALL	127.0.0.1/32	

2. The list of SG Inbound/Outbound rules created “Stack-PCSVIntSG” are:

Figure 62: Stack-PCSVIntSG - Inbound Rules

sg-5620822f | sgssgilStack-PCSVIntSG

Summary Inbound Rules Outbound Rules Tags

Edit

Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP (6)	6667	0.0.0.0/0	
All ICMP - IPv4	ICMP (1)	ALL	0.0.0.0/0	

Figure 63: Stack-PCSVIntSG - Outbound Rules

sg-5620822f | sgssgilStack-PCSVIntSG

Summary Inbound Rules **Outbound Rules** Tags

Edit

Type	Protocol	Port Range	Destination	Description
ALL Traffic	ALL	ALL	0.0.0.0/0	

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

Figure 64: Stack-PCSVMgmtSG - Inbound Rules

sg-be2183c7 | sgssgilStack-PCSVMgmtSG

Summary **Inbound Rules** Outbound Rules Tags

Edit

Type	Protocol	Port Range	Source	Description
HTTP (80)	TCP (6)	80	0.0.0.0/0	
Custom TCP Rule	TCP (6)	6667	0.0.0.0/0	
Custom TCP Rule	TCP (6)	830	0.0.0.0/0	
HTTPS (443)	TCP (6)	443	0.0.0.0/0	
All ICMP - IPv4	ICMP (1)	ALL	0.0.0.0/0	

Figure 65: Stack-PCSVMgmtSG - Outbound Rules

sg-be2183c7 | sgssgilStack-PCSVMgmtSG

Summary Inbound Rules **Outbound Rules** Tags

Edit

Type	Protocol	Port Range	Destination	Description
ALL Traffic	ALL	ALL	127.0.0.1/32	

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 <https://www.pulsesecure.net> 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](#).

```
<pulse-config><primary-dns>8.8.8</primary-dns><secondary-dns>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>val.psecure.net</cert-common-name><cert-random-text>{d4f1p0nysfms}</cert-random-text><cert-organisation>Psecure</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 External 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 <https://www.pulsesecure.net> 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	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 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.

```
"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](#).

```
<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>val.psecure.net</cert-common-name><cert-random-text>defpisonysfms</cert-random-text><cert-organisation>Psecure Qxx</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>
```

VPCID: This is the ID of the existing VPC.

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

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

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

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

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

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

```
"SubnetMgmtId" : {
  "Type" : "String",
  "Description" : "ID of the subnet where PCS Management 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 External 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>.