

Pulse Policy Secure Virtual Appliance on Amazon Web Services

Deployment Guide

Published Date

November 2020

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 Policy 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
1.2 November 2020	PPS in AWS Marketplace	
1.1 September 2020	Accessing the Pulse Policy Secure Virtual Appliance as an Administrator section is updated with PPS IP and credentials.	
1.0 March 2020	First version of the document.	
1.1 September 2020	Updated the "Accessing the Pulse Policy Secure Virtual Appliance as an Administrator" section with PPS url and default password.	

Table of Contents

Revision History	3
Overview	6
About This Guide	6
Assumptions	6
Pulse Policy Secure in AWS Marketplace	6
Prerequisites and System Requirements on AWS Marketplace	6
Deploying Pulse Policy Secure on AWS Marketplace	7
Select Template	9
Specify Details	10
Review	11
Pulse Policy Secure on Amazon Web Services	12
Prerequisites and System Requirements on AWS	12
Deploying Pulse Policy Secure on Amazon Web Services	12
Supported Platform Systems	13
Steps to Deploy Pulse Policy Secure on AWS	13
Registering the AMI	14
Prerequisites	14
Deploying Pulse Policy Secure on AWS using AWS Portal	15
Deploying PPS on New Virtual Private Cloud	15
Deployment on VM with Three NIC Cards	15
Deployment on VM with Two NIC Cards	
Deploying PPS on an Existing Virtual Private Cloud	20
Deployment on VM with Three NIC Cards	20
Deployment on VM with Two NIC Cards	23
Pulse Policy Secure Provisioning Parameters	24
Configuring Licenses on the Pulse Policy Secure Appliance	25
Pulse License Server in Corporate Network	25
Pulse License Server in Cloud Network	26
Adding Authentication Code in PPS Admin Console	27
Including Authentication Code in CloudFormation Template	27
Accessing the Pulse Policy Secure Virtual Appliance	27
Accessing the Pulse Policy Secure Virtual Appliance as an Administrator	
Accessing the Pulse Policy Secure Virtual Appliance as an End User	
Accessing the Pulse Policy Secure Virtual Appliance using SSH Console	29
On Linux and Mac OSX	29
On Windows	29
System Operations	
Network Configuration	
IP Address Assignment for Internal, External and Management Interfaces	
IP Addressing Modes	31
Modifying Network Parameters After Deployment	

Controlling the Selection of Internal, External and Management Interfaces	
Backing up Configs and Archived Logs on S3 Bucket	
Configuring Backup Configs and Archived Logs via PPS Admin Console	
Configuring Backup Configs and Archived Logs via REST	
Setting AWS as Archive Logs Backup	
Decommissioning Pulse Policy Secure	35
Pricing	
Limitations	
Troubleshooting	
Frequently Asked Questions	
Appendix A: Security Group (SG)	
Appendix B: Pulse Policy Secure CloudFormation Template	42
Parameters	42
Resources	44
Outputs	46
Appendix C: Pulse Policy Secure CloudFormation Template for an Existing Virtual Private Cloud	47
Parameters	47
Resources	49
Outputs	50
References	51
Requesting Technical Support	51

Overview

About This Guide

This guide helps in deploying the Pulse Policy Secure Virtual Appliance on Amazon Web Services (AWS). A Pulse Policy Secure administrator can manually upload the Pulse Policy Secure Virtual Appliance image (AMI) into AWS storage account. Once the AMI package is available in the AWS storage account, the Pulse Policy Secure administrator can deploy Pulse Policy Secure on AWS in the cloud.

Assumptions

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

Pulse Policy Secure in AWS Marketplace

Pulse Policy Secure is made available in AWS Market Place. The CloudFormation templates are available at <u>Amazon marketplace</u>.

Prerequisites and System Requirements on AWS Marketplace

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

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

Deploying Pulse Policy Secure on AWS Marketplace

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

Figure 1: AWS Marketplace

👯 aws r	narketplace				Pulse Policy Secure		
Categories 👻	Delivery Methods 👻	Solutions 👻	AWS IQ 👻	Your Saved List			
				Categories	Pulse Policy Se	ecure (5 results) showing 1 - 5	
				All Categories Infrastructure Software DevOps (4)	: (5) SPulse Secure:	Pulse Policy Secure - BYOL 2 NIC Version 9.1R8 Sold by Pulse Secure LLC	
				Industries (1) Filters		Pulse Policy Secure (PPS) is a next-gen NAC that enables organizations to gain complete visibility, understand their security posture, and enforce roles-based access and endpoint security policy for	
				Vendors Pulse Secure LLC (5)		network user, guest and IoT devices. Leveraging core network, mobile and security infrastructure Linux/Unix, CentOS 6.4 - 64-bit Amazon Machine Image (AMI)	
				Operating System	S Pulse Secure	Pulse Policy Secure - BYOL 3 NIC Version 9.1R8 Sold by Pulse Secure LLC	
				Bring Your Own Licen	ise (4)	Pulse Policy Secure (PPS) is a next-gen NAC that enables organizations to gain complete visibility, understand their security posture, and enforce roles-based access and endpoint security policy for network user, guest and IoT devices. Leveraging core network, mobile and security infrastructure	
				Delivery Method		Linux/Unix, CentOS 6.4 - 64-bit Amazon Machine Image (AMI)	

AWS Marketplace contains the following two Pulse Policy Secure SKUs:

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

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

👯 aws r	marketplac <u>e</u>							Q	
ategories 👻	Delivery Methods 👻	Solutions 👻	AWS IQ 👻	Your Saved List					Parti
					Pulse Policy Secure - BYOL	3 NIC		Continue to Subscribe	
				Secure [®]	By: Pulse Secure LLC C Latest Version: 9.1	R8		Save to List	
					Linux/Unix 0 AWS reviews	prise apps and serv	vices in the data centre or cloud.	Typical Total Price \$0.199/hr Total pricing per instance for services hosted on c4.xlarge in US East (N. Virginia). View Details	
				Overview	Pricing	Usage	Support	Reviews	
				Pulse Policy Secure (PPS) complete visibility, under access and endpoint secu Leveraging core network, NAC solution can streaml onboarding and IoT secur malware, rogue device, un Pulse Policy Secure inclue NAC as well as VPN offer single, integrated, remote and dynamic VPN feature well as standards-based I Management options incl well as a centralized man appliance - Pulse One.	is a next-gen NAC that enables organizations to c stand their security posture, and enforce roles-ba rity policy for network user, guest and IoT devices mobile and security infrastructure integrations, P ine endpoint compliance and remediation, BYOD ity, as well as automate threat response to mitiga nauthorized access and data leakage risks. des Pulse Secure Clients that are universal clients i ngs from Pulse Secure, reducing complexity with a access client that can also provide LAN access co is to remote users. Pulse Clients upport SSL, ESP PSEC support for mobile devices and IoT. uude a web-based intuitive UI, XMLRPC and REST , agement console available in cloud or as an on-pr	iain sed H	ighlights Pulse Policy Secure enables secure device to enterprise apps and servic center or cloud. Pulse Policy Secure (PPS) is a next- enables organizations to gain comp understand their security posture, a based access and endpoint security user, guest and IoT devices Ease of Administration: Managemen web-based intuitive UI, XMLRPC an as a centralized management consc or as an on-prem appliance - Pulse	access from any tess in the data gen NAC that plete visibility, and enforce roles- policy for network nt options include a d REST APIs as well ole available in cloud One.	

- 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. Under Delivery Method, select either Existing VPC or New VPC that you want to deploy and click **Continue to Launch**.

Figure	-igure 3: Launch CloudFormation									
👯 aws r	narketplace				Q					
Categories 👻	Delivery Methods 👻	Solutions 👻	AWS IQ 👻	Your Saved List	F					
				Secure Pulse Policy Secure - BYOL 3 NIC	Continue to Launch					
				< Product Detail Subscribe <u>Configure</u>						
				Configure this software	Pricing information					
				Choose a fulfillment option below to select how you wish to deploy the software, then enter the information required to configure the deployment.	This is an estimate of typical software and infrastructure costs based on your configuration. Your actual charges for each statement nericd may differ from					
				Delivery Method	this estimate.					
				Pulse Policy Secure - 3 nic w/new VPC	Pulse Policy \$0/hr					
				Select a CloudFormation template	NIC BYOL running on					
				Pulse Policy Secure - 3 nic w/existing VPC	e interge					
				Pulse Policy Secure - 3 nic w/new VPC						

5. Select the Software Version and the Region to deploy the software.

🐓 aws r	marketplace				
Categories 👻	Delivery Methods 👻	Solutions 👻	AWS IQ 👻	Your Saved List	
	Delivery Met Amazon Machine I Amazon SageMak	hods mage er		Pulse Policy Secure - BYOL 3 NIC	
	AWS Data Exchang CloudFormation S Container Private Image Buil SaaS	je tack d		< Product Detail Sub Configure Choose a fulfillment required to configur Delivery Method Pulse Policy Secu Software Version 9.1R8 (Nov 20, 24 Whats in T Pulse Policy Region US East (N. Virgin Use of Local Zones Product code: 47ms Release notes (upda	scribe <u>Configure</u> this software to ption below to select how you wish to deploy the software, then enter the information the deployment.

In the Launch page displayed, select Launch CloudFormation and click Launch.

👯 aws i	narketplace	Solutions 💌	AWS IO 👻	Your Saved List		Q
				Secure:	Pulse Policy Secure - BYOL 3 NIC	
				< Product Detail Sub- Configuration De Fulfillment Option Region Usage Instruct	scribe Configure Launch IS SOFtWare Intation and choose how you wish to launch the software. etails Pulse Policy Secure - 3 nic w/new VPC Pulse Policy Secure - 8 YOL 3 NIC running on c4.storge 9.1R8 US East (N. Virginia)	
				Choose Action	mation Choose this action to launch your configuration through the AWS CloudFormation console.	

Select Template

6. 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 template	Create stack
Step 2 Specify stack details	Prerequisite - Prepare template
Step 3 Configure stack options	Prepare template Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack
Step 4 Review	
	A template is a JSON or YAML file that describes your stack's resources and properties.
	Selecting a template generates an Amazon S3 URL where it will be stored. Amazon S3 URL Upload a template file
	Amazon S3 URL
	Amazon S3 URL https://s3.amazonaws.com/awsmp-fulfillment-cf-templates-prod/4728c58d-efdc-4cc9-9ad2-f6b2a60987c9.1e238774-b5d7-4008-b035-004

Specify Details

- 7. In the Specify Stack Details page, specify a name for the stack.
- 8. In the Parameters section, use the default parameter values. These are defined in the CloudFormation template.
- 9. In the Pulse Policy Secure Configuration section:
 - Select Pulse Policy Secure VM size. Default: t2.medium
 - PPS admin username is configured by default. Default: ppsadmin. 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 Policy Secure Provisioning Parameters.
 - Select SSH Key Name of EC2 key pair. This key is used to access PPS via SSH. The SSH keys are generated using ssh-keygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer <u>http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-keypairs.html.</u>

Figure 5: Specify Configuration Details

CloudFormation > Stacks >	Create stack
Step 1 Specify template	Specify stack details
Step 2 Specify stack details	Stack name
	Stack name
Step 3	pps-stack
Configure stack options	Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).
Step 4	
Review	Parameters
	Parameters are defined in your template and allow you to input custom values when you create or update a stack.
	New VPC Configuration
	CIDR block for entire VPC.
	10.20.0.0/16
	Internal Subnet address space Pulse Policy Secure VM Internal Interface connects to this subnet
	10.20.1.0/24
	External Subnet address space Pulse Policy Secure VM external interface connects to this subnet
	10.20.2.0/24
	Management Subnet address space Pulse Policy Secure VM management interface connects to this subnet
	10.20.3.0/24
	Pulse Policy Secure Configuration
	Instance Type
	t2.medium
	Admin User Name
	ppsadmin
	Admin Descured
	Passivo association of the Pulse Policy Secure admin user, should be minimum 10 characters
	Config Data Pulse Policy Secure configuration data.
	<pre></pre>
	SSH Key Name

Review

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

lo	pudFormation > Stacks > Create stack
Q	uick create stack
	Template
	Template IIPI
	https://s3.amazonaws.com/awsmp-fulfillment-cf-templates-prod/4728c58d-efdc-4cc9-9ad2-f6b2a60987c9.1e238774-b5d7-4008-b035-
	00ed8dae78a2.template
	Stack description
	Pulse Policy Secure with three interfaces deployed on new VPC
	Stack name
	Stack name
	pps-stack
	Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).
	Parameters
	Parameters are defined in your template and allow you to input custom values when you create or update a stack.
	New VPC Configuration
	New VPC address space
	CIDR block for entire VPC.
	10.20.00/16
	Internal Subnet address space Pulse Policy Secure VM internal interface connects to this subnet
	10.20.1.0/24
	External Subnet address space
	Pulse Policy Secure VM external interface connects to this subnet
	10.20.2.0/24
	Management Subnet address space
	10.20.3.0/24
	Pulse Policy Secure Configuration
	Instance Type
	Pulse Policy Secure instance type t 2 medium
	Sector Se
	Admin User Name Pulse Policy Secure admin user.
	ppsadmin
	Admin Dassword
	Admin Password Password for the Pulse Policy Secure admin user, should be minimum 10 characters
	Config Data Pulse Policy Secure configuration data.
	<pre></pre>
	SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair
	sachineast

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

av	/S Services ▼							¢ *	N. Virginia 🔻 Support 🔻
	CloudFormation > Stacks > pps-stack								
	🗉 Stacks (11) 🛛 🖒	pps-stack					Delete Upda	ate Stack actions	Create stack V
	Q Filter by stack name	Stack info Events Reso	ources Outputs	Parameters Templ	ate Change sets				
	Active View nested								
		Events (1)							C
	2020-11-23 16:28:58 UTC+0530 CREATE_IN_PROGRESS	Q Search events							۲
	sachinmarket	Timestamp		Logical ID	Status		Status reason		
	2020-11-21 14:03:25 UTC+0530 ROLLBACK_COMPLETE	2020-11-23 16:28:58 UTC+0530		pps-stack	CREATE_I	N_PROGRESS	User Initiated		
	PZTA-AG 2020-09-04 10:15:55 UTC+0530 ⊘ CREATE_COMPLETE								

To access Pulse Policy Secure Virtual Appliance, see Accessing the Pulse Policy Secure Virtual Appliance

Pulse Policy Secure on Amazon Web Services

Prerequisites and System Requirements on AWS

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

- AWS account
- Access to the AWS portal (<u>https://console.aws.amazon.com/</u>)*
- Pulse Policy Secure Virtual Appliance Image (.ami file)
- AWS CloudFormation template
- Pulse Policy Secure licenses **
- Site-to-Site VPN between AWS and the corporate network (optional)

Note: This is needed only if the Pulse Policy Secure users need to access corporate resources.

- Pulse License Server (optional)**
 - o Located at corporate network, accessible through site-to-site VPN
- Pulse Policy Secure configuration in XML format (optional)

note:

* Pulse Policy Secure Virtual Appliance can be deployed only through AWS CloudFormation style. ** From 9.0R3 release, Pulse Policy Secure Virtual Appliance, by default, has two evaluation licenses, and supports licensing with License server located at corporate network.

Deploying Pulse Policy Secure on Amazon Web Services

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

Figure 6: Pulse Policy Secure on AWS



Supported Platform Systems

This section helps you in choosing the instance types that should be deployed with Pulse Policy 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.medium	2	24	4	EBS-Only
t2.large	2	36	8	EBS-Only
t2.xlarge	4	54	16	EBS-Only

Steps to Deploy Pulse Policy Secure on AWS

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

• Registering the AMI

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

• Deploying Pulse Policy 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 PPS 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 <u>https://aws.amazon.com/cli/</u>.
- 3. Copy PPS Xen image on the client machine.
- 4. Create Amazon S3 bucket and VM Import service role by following the procedures mentioned in <u>https://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-image-import.html#vmimport-iam-permissions</u>
- Upload the PPS 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-devicemappings DeviceName="/dev/xvda",Ebs={SnapshotId=<snapshot-id>} --virtualization-type hvm --root-device-name "/dev/xvda" --region <your-ec2-region>

 Once the snapshot is created, you can also copy the snapshot ID from the AWS console (Services > EC2 > Elastic Block Store > Snapshots). Select the snapshot and click Actions > Create Image. This completes AMI registration.

	Na	PSA-V-XEN-P	PS-3701.1-SEF	RIAL-xen.img	Descrip	tion	PSAv12	346]	
	Architecture	(j) (x86_64		4	Virtualization type	1	Hardwa	re-assisted virtu	alization 🔹	Ĩ.	
Root	device name	(j) /dev/xvda			Kernel ID	(i)	Use def	ault	4		
-1	RAM disk ID	() Use default		4							
Block	Device Mappi	ngs									
plume pe	Device	Snapshot (j)	Size (GiB)	Volume Type	• ①		IOPS (j)	Throughput (MB/s) (i)	Delete on Termination	Encrypted	
unt.	/dev/xvda	snap-	40	General Pur	oose SSD (gp2)	~	120 / 3000	N/A			Not Encrypt

Deploying Pulse Policy 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 Policy Secure deployment.

Pulse Policy Secure can be deployed:

- on <u>a new Virtual Private Cloud</u> or
- on an already existing Virtual Private Cloud
- as <u>a license server</u>

Deploying PPS on New Virtual Private Cloud

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

Deployment on VM with Three NIC Cards

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

1. Select AWS Services > CloudFormation and click Create stack > With new resources (standard).

Figure 7: Create New Stack

aws	Services 🗸	Resource Groups 🐱	*		¢		N. California 👻	Support
CloudFo	ormation > Stacks	5						
Sta	cks (56)) Delete	Update	Stack actions	Create sta	ack 🔺
Q	Filter by stack name	2		Active 🔻	View nester	d With new resource With existing reso	es (standard) urces (import reso	urces)

2. Select **Upload a template file**. Click **Choose File** and select "pulsesecure-pps-3-nics-new-network.json" template file for the new VPC. Click **Next**.

Figure 8: Upload Template

Step 1 Specify template	Create stack
Step 2 Specify stack details	Prerequisite - Prepare template
Step 3 Configure stack options	Prepare template Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to inclu in the stack. The template is ready Duse a sample template Create template in Designer
Step 4	
leview	
leview	Specify template A template is a JSON or YAML file that describes your stack's resources and properties.
leview	Specify template A template is a JSON or YAML file that describes your stack's resources and properties. Template source Selecting a template generates an Amazon S3 URL where it will be stored.
leview	Specify template A template is a JSON or YAML file that describes your stack's resources and properties. Template source Selecting a template generates an Amazon S3 URL where it will be stored. Amazon S3 URL
leview	Specify template A template is a JSON or YAML file that describes your stack's resources and properties. Template source Selecting a template generates an Amazon S3 URL where it will be stored. Amazon S3 URL Upload a template file Choose file 3nic%20newvpc.txt JSON or YAML formatted file

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

(i) Note: Before proceeding with deployment, ensure that the attribute "accept-license-agreement" in PPSConfigData is set to "y".

Figure 9: Specify Details for New Virtual Private Cloud

aws	Services 🗸	Resource Groups 🐱	*		¢	*	N. California 💌	Support
CloudFo	ormation > Stack	<s> Create stack</s>						
Step 1 Specify	template	Specif	y stack deta	iils				
Step 2 Specify	stack details	Stack r	iame					
Step 3 Configu	ire stack options	Stack nam Enter a Stack nam	ne stack name e can include letters (A-Z ar	rd a-z), numbers (0-9), and dashes (-).				
Step 4 Review		Parame	e ters s are defined in your templa	ate and allow you to input custom values wh	en you create	or update a stack.		
		New VPC New VPC CIDR block	C Configuration address space for entire VPC.					
		10.200	0.0/16					
		Internal S PPS intern 10.200	al interface connects to this 11.0/24	subnet				
		External PPS extern	Subnet address space al interface connects to this	s subnet				
		10.200	12.0/24					
		Managen PPS manag	ient Subnet address spa jement interface connects t 	ace to this subnet				
		PPS Con	figuration					
		PPS AMI AMI ID of y	D our existing PPS image					
		Instance Select PPS	Type instance type					
		t2.med	ium					▼
		PPS Confi PPS config	ig Data data	111.				
		SSH Key	conпg> <wins-server>1. Name</wins-server>	1.1.1 <dns-domain>pse</dns-domain>	cure.net <td>ans-αomain><admin-us< td=""><td>ername>admin<td>aamin-us</td></td></admin-us<></td>	ans-αomain> <admin-us< td=""><td>ername>admin<td>aamin-us</td></td></admin-us<>	ername>admin <td>aamin-us</td>	aamin-us
		Name of a	1 existing EC2 KeyPair. Your	PP's will launch with this KeyPair.				▼
						Cancel	Previous	Next

- **Stack name**: Specify the stack name in which Pulse Policy Secure needs to be deployed
- New VPC address space: Virtual private cloud address space
- Internal Subnet address space: Subnet from which Pulse Policy Secure internal interface needs to lease IP
- External Subnet address space: Subnet from which Pulse Policy Secure external interface needs to lease IP
- Management Subnet address space: Subnet from which Pulse Policy Secure management interface needs to lease IP
- PPS AMI ID: ID of the uploaded AMI file
- Instance Type: Size of the instance t2.medium or t2.xlarge or t2.2xlarge.
- **PPS Config Data**: Provisioning parameters in an XML format. For details, see Pulse Policy Secure Provisioning Parameters.
- SSH Key Name: This key is used to access PPS via SSH. The SSH keys are generated using sshkeygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer <u>http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html.</u>

4. Click Next. Review the specified details and click **Create Stack stack**. Observe the deployed PPS in a few minutes.

-			
aws Services -	Resource Groups 🗸 🛧	凢 sachin @ pulseqa × N. California ×	Support +
CloudFormation \times	CloudFormation > Stacks > PPS3NIC-1		
Stacks	E Stacks (53)	PPS3NIC-1 Delete Update Stack actions V Cre	ate stack 🔻
Stack details Drifts	Q. Filter by stack name	Stack info Events Resources Outputs Parameters Template Change sets	
StackSets Exports	Active View nested	Events (74)	C
Designer	PP52NIC-1 2020-02-26 16:09:46 UTC+0530 ✓ CREATE_COMPLETE	Q. Search events	۲
	PPS3NIC-1	Timestamp 🔻 Logical ID Status Status reason	
 CloudFormation registry Resource types 	2020-02-26 16:03:57 UTC+0530 ⊘ CREATE_COMPLETE	2020-02-26 16:06:12 UTC+0530 PPS3NIC-1 OCREATE_COMPLETE -	
	PPS3NIC	2020-02-26 16:06:10 UTC+0530 EC2Instance OCREATE_COMPLETE -	
	2020-02-26 15:19:31 UTC+0530	2020-02-26 16:05:17 UTC+0530 EC2Instance OCREATE_IN_PROGRESS Resource creation Initiated	
	C ROLLBROK_COMPLETE	2020-02-26 16:05:15 UTC+0530 EC2Instance O CREATE_IN_PROGRESS -	
Previous console	awsgw840 O 2020-02-25 17:26:19 UTC+0530	2020-02-26 16:05:13 UTC+0530 EIPAssoc1 OC CREATE_COMPLETE -	
Feedback	CREATE_COMPLETE	2020-02-26 16:05:13 UTC+0530 EIPAssoc0 OCREATE_COMPLETE -	

Figure 10: New VPC

Deployment on VM with Two NIC Cards

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

1. Select AWS Services > CloudFormation and click Create stack > With new resources (standard).

Figure 11: Create New Stack

aws	Services 🗸	Resource Groups	× 1:			Ą	······································	N. California 👻	Support
CloudF	ormation > Sta	cks							
Sta	acks (56)) C	Delete	Update	Stack actions	Create st	ack 🔺
Q	Filter by stack na	me		Active	▼	View neste	d With new resource With existing reso	es (standard) ources (import resi	ources)

2. Select **Upload a template file**. Click **Choose file** and select "pulsesecure-pps-2-nics-new-network.json" template file for the new VPC. Click **Next**.

Figure 12: Upload Template

Prerequisite - Prepare template Prepare template	o 1 ecify template	Create stack
Prepare template p 3 frigure stack options P 4 iew P 2 iew Template is ready Use a sample template Create template in Designer Specify template template is a JSON or YAML file that describes your stack's resources and properties. Template source Selecting a template generates an Amazon S3 URL where it will be stored. Implate is a upplate file Upload a template file Implate is a upplate file	p 2 ecify stack details	Prerequisite - Prepare template
Create template is ready Use a sample template Create template in Designer	p 3 nfigure stack options	Prepare template Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.
Selecting a template generates an Amazon 53 URL where it will be stored.	2p 4	
Upload a template file Choose file A 2nic%20newvpc.txt	view	Specify template A template is a JSON or YAML file that describes your stack's resources and properties. Template source
550 v of Mele formatice file	view	Specify template A template is a JSON or YAML file that describes your stack's resources and properties. Template source Selecting a template generates an Amazon S3 URL where it will be stored. O Amazon S3 URL

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 PPSConfigData is set to "y".

Figure 13: Specify Details for New Virtual Private Cloud

/S Services ~ Reso	urce Groups 🗸 🔭
CloudFormation > Stacks >	Create stack
Specify template	Specify stack details
itep 2 Specify stack details	Stack name
	Stack name
itep 3 Configure stack options	PPS2NIC-1
j	Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).
itep 4	
(eview	Parameters Parameters are defined in your template and allow you to input custom values when you create or update a stack.
	New VPC Configuration
	New VPC address space CIDR block for entire VPC.
	10.200.0.0/16
	Internal Subnet address space PPS internal interface connects to this subnet
	10.200.11.0/24
	External Subnet address space PPS external interface connects to this subnet
	10.200.12.0/24
	PPS Configuration
	PPS AMI ID AMI ID of your existing PDS impos
	ami-032b0bf4376b91cbd
	Instance Type
	t2.medium
	PPS Config Data
	<pre></pre> <pre></pre> <pre></pre> <pre></pre>
	SSH Key Name Name of an existing EC2 KeyPair, Your PPS will launch with this KeyPair.
	sachin-latest
	Cancel Previous Next

- Stack name: Specify the stack name in which Pulse Policy Secure needs to be deployed
- New VPC address space: Virtual private cloud address space
- Internal Subnet address space: Subnet from which Pulse Policy Secure internal interface needs to lease IP
- External Subnet address space: Subnet from which Pulse Policy Secure external interface needs to lease IP
- PPS AMI ID: ID of the uploaded AMI file
- Instance Type: Size of the instance t2.medium or t2.large
- **PPS Config Data**: Provisioning parameters in an XML format. For details, see Pulse Policy Secure Provisioning Parameters.

- SSH Key Name: This key is used to access PPS via SSH. The SSH keys are generated using sshkeygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer <u>http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html.</u>
- 4. Review the specified details and click **Create stack**. Observe the deployed PPS in a few minutes.

Figure 14: New VPC

aws Services ~	Resource Groups 🗸 🔸		ậ sachin @ pulseqa ▾ N. California ▾ Support ▾
CloudFormation \times	CloudFormation > Stacks > PPS2NIC-1		
Stacks	🖸 Stacks (53)	PPS2NIC-1	update Stack actions ▼ Create stack ▼
Drifts	Q Filter by stack name	Stack info Events Resources Outputs Parameters Template Change sets	
StackSets Exports	Active View nested < 1 >	Events (56)	
Designer	PP52NIC-1 2020-02-26 16:09:46 UTC+0530 ⊘ CREATE_COMPLETE	Q Search events	©
CloudFormation registry	PPS3NIC-1	Timestamp V Logical ID Status	Status reason
Resource types	2020-02-26 16:03:57 UTC+0530	2020-02-26 16:11:40 UTC+0530 PPS2NIC-1 OCREATE_COMPLETE	-
	PPS3NIC	2020-02-26 16:11:38 UTC+0530 EC2Instance OCREATE_COMPLETE	
	2020-02-26 15:19:31 UTC+0530	2020-02-26 16:11:06 UTC+0530 EC2Instance ③ CREATE_IN_PROGRESS	Resource creation Initiated
Drouious concelo	awsgw840	2020-02-26 16:11:04 UTC+0530 EC2Instance ③ CREATE_IN_PROGRESS	

Deploying PPS on an Existing Virtual Private Cloud

This section describes PPS deployment with <u>three NIC cards</u> and <u>two NIC cards</u>.

Deployment on VM with Three NIC Cards

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

1. Select AWS Services > CloudFormation and click Create stack > With new resources (standard).

Figure 15: Create New Stack

aws	Services 🗸	Resource Groups 🗸	* *	۵	••••••• •• ••••••• •	N. California 👻	Support
CloudFor	mation > Stack	5					
Stac	ks (56)			J Delete Update	Stack actions	Create sta	ack 🔺
Q	Filter by stack nam	e		Active View net	With new resour With existing res	ces (standard) ources (import reso	ources)

2. Select **Upload a template file.** Click **Choose file** and select "pulsesecure-pps-3-nics-existing-vpc.json" template file for existing VPC. Click **Next**.

Figure 16: Upload Template

esource Groups 🗸 🔭 N. California 🖌 Suppo
Create stack
Prerequisite - Prepare template
Prepare template Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack. Image: Image
Specify template A template is a JSON or VAML file that describes your stack's resources and properties.
Template source Selecting a template generates an Amazon S3 URL where it will be stored.
Amazon S3 URL Upload a template file
Upload a template file Choose file JSON or YAML formatted file
S3 URL: https://s3-us-west-1.amazonaws.com/cf-templates-udohotbi8ez9-us-west-1/2020064kRF-3nic%20exsiting%20vpc.t View in

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

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

Figure 17: Specify Details for Existing Virtual Private Cloud

	Resource Groups 🗸 🏠 N. California 👻 Suppo
udFormation > Stack	KS 📏 Create stack
1 cify template	Specify stack details
2 cify stack details	Stack name
	Stack name
3 figure stack options	3NICexisting
	Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).
4	
lew	Parameters Parameters are defined in your template and allow you to input custom values when you create or update a stack.
	Existing VPC details Existing VPC ID
	ID of existing VPC vpc-0291096508668c9fe
	Internal Subnet ID ID of the subnet where PPS internal interface connects
	subnet-0b56e14aa22ee9463
	External Subnet ID ID of the subnet where PPS External interface connects
	subnet-0b177718d594cd027
	Management Subnet ID ID of the subnet where PPS Management interface connects
	subnet-01a7588fd06e8a4fe
	PPS Configuration
	AMI ID of your existing PPS image
	ami-032b0bf4376b91cbd
	Instance Type Select PPS instance type
	t2.medium
	PPS Config Data PPS config data
	<pre><pulse-config><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net</dns-domain><admin-username>admin</admin-username>adminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadmin</pulse-config></pre>
	SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.

- Stack name: Specify the stack name in which Pulse Policy Secure needs to be deployed
- Existing VPC ID: Virtual private cloud ID
- Internal Subnet ID: Subnet from which Pulse Policy Secure internal interface needs to lease IP
- External Subnet ID: Subnet from which Pulse Policy Secure external interface needs to lease IP
- Management Subnet ID: Subnet from which Pulse Policy Secure management interface needs to lease IP
- PPS AMI ID: ID of the uploaded AMI file
- Instance Type: Size of the instance t2.medium or t2.large
- **PPS Config Data**: Provisioning parameters in an XML format. For details, see Pulse Policy Secure Provisioning Parameters.
- SSH Key Name: This key is used to access PPS via SSH. The SSH keys are generated using sshkeygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer <u>http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html</u>
- 4. Review the specified details and click **Create stack**. Observe the deployed PPS in a few minutes.

Deployment on VM with Two NIC Cards

To deploy Pulse Policy 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

aws	Services 🗸	Resource Groups 🐱	*	۵		N. California 👻	Support
CloudFor	mation > Stack	5					
Stac	ks (56)			Delete Update	Stack actions 🔻	Create sta	ick 🔺
Q	Filter by stack nam	e		Active View nested	With new resource With existing reso	s (standard) urces (import reso	urces)

- 2. Select **Upload a template to Amazon S3**. Click **Browse** and select "pulsesecure-PPS-2-nics-existing-vpc.json" template file for existing VPC. Click **Next**.
- 3. In the Specify Stack Details page, fill or modify the following parameters.

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

Figure 19: Specify Details for Existing Virtual Private Cloud

stransformation & back & centents of comparison of stransformation & back & centents of stransformation & centents of st	S Services - Res	source Groups 🗸 🔺	Д • 4	······································	N. California 👻	Support 🗸
sh and show a data data a data a data a data a data data data a data	loudFormation > Stacks >	Create stack				
star Stark name singures stark options Stark name stark store can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and dathers (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and and and (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and (A). Stark Name can include latters (A: 2 and a-2, numbers (B-9), and (A). Stark Name can	ep 1 pecify template	Specify stack details				
Size A composition of the second of the seco	ep 2 becify stack details	Stack name				
exiting2nic Stark runne can include letters (A.2 and a-2, numbers (0-9), and dashes (-). Stark nume can include letters (A.2 and a-2, numbers (0-9), and dashes (-). Parameters		Stack name				
Stack name can include letters (A-2 and a-2), numbers (B-9), and dashes (-). Stack name can include letters (A-2 and a-2), numbers (B-9), and dashes (-). Parameters are defined in your template and allow you to input custom values when you create or update a stack. New VPC Configuration New VP	p 3 pfigure stack options	existing2nic				
Parameters		Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).				
Parameters Parameters are defined in your template and allow you to input custom values when you create or update a stack. New VPC Configuration New VPC address space 10.200.00/16 Internal Subnet address space PPS internal interface connects to this ubnet 10.200.11.0/24 PPS Configuration PPS Configuratio	p 4 view					
New VPC Configuration LNB block for entite VPC. 10.200.00/16 10.200.00/16 PPS internal Subnet address space PPS internal Subnet address space PPS configuration PPS Configuration PPS Configuration PPS Add ID Add ID of your existing PPS image Instance Type Select PPS instance type tzmeelium PPS Config Data PPS config vwins-server>1.1.1.1 <dns-domain>psecure.net PPS Config Cata *putse-config> Mate an existing EC2 KeyPair. Your PPS will launch with this KeyPair.</dns-domain>		Parameters Parameters are defined in your template and allow you to input custom values when you create or update a	ı stack.			
New VPC address space CIDR block for entire VPC. 10.200.00/16 Internal Subnet address space PPS internal Interface connects to this subnet 10.200.11.0/24 External Subnet address space PPS ortinguartion PPS Configuration PPS AMI ID AMI ID or your existing PPS image ct.mending up to respece the space PPS Configuration PPS Configuration PPS Config Data Select PPS instance type		New VPC Configuration				
10.200.0.0/16 Internal Subnet address space PPS internal interface connects to this subnet 10.200.11.0/24 External Subnet address space PPS configuration PPS SMI ID AMI ID of your existing PPS image Instance Type Select PPS instance type 12.20eding Data PPS configo Data PPS configo Taba Cyclie-configo Cata PPS configo Taba Select RPS instance type Instance Type PPS configo Taba Select RPS Nem Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.		New VPC address space CIDR block for entire VPC.				
Internal Subnet address space PPS internal interface connects to this subnet 10.200.11.0/24 External Subnet address space PPS external interface connects to this subnet 10.200.12.0/24 PPS Configuration PPS Add ID MI ID of your existing PPS image		10.200.0.0/16				
10.200.11.0/24 External Subnet address space PPS external interface connects to this subnet 10.200.12.0/24 PPS Configuration PPS MI ID AMI D of your existing PPS image Instance Type Select PPS instance type t2.medium PPS Config Data PPS config Data PPS config Data PPS config Cata SHE Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.		Internal Subnet address space PPS internal interface connects to this subnet				
External Subnet address space PPS external interface connects to this subnet 10.200.12.0/24 PPS Configuration PPS AMI ID AMI ID of your existing PPS image Instance Type Select PPS instance type Select PPS instance type 2		10.200.11.0/24				
10.200.12.0/24 PPS Configuration PPS AMI ID AMI ID of your existing PPS image Instance Type Select PPS instance type t2.medium PPS Config Data PPS config Data PPS config Data <public-config><wins-server>1.1.1.1 SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.</wins-server></public-config>		External Subnet address space PPS external Interface connects to this subnet				
PPS Configuration PPS AMI ID AMI ID of your existing PPS image Instance Type Select PPS instance type t2.medium PPS Config Data PPS config Data Cypulse-config>swins-server>1.1.1.1 SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.		10.200.12.0/24				
PPS AMI ID AMI ID of your existing PPS image Instance Type Select PPS instance type t2.medium PPS Config Data PPS config Data Cypulse-config>swins-server>1.1.1.1 SH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.		PPS Configuration				
Instance Type Select PPS instance type t2.medium PPS Config Data PPS config data <pulse-config><wins-server>1.1.1.1 SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.</wins-server></pulse-config>		PPS AMI ID AMI ID of your existing PPS image				
t2.medium PPS Config Data PPS config data <pulse-config><wins-server>1.1.1.1 SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.</wins-server></pulse-config>		Instance Type Select PPS instance type				
PPS config Data PPS config data <putse-config><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net</dns-domain><admin-username>admin</admin-username><admin-1< td=""> SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.</admin-1<></putse-config>		t2.medium				•
<pre><pulse-config><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net</dns-domain><admin-username>admin</admin-username><admin-] <="" an="" ec2="" existing="" key="" keypair.="" launch="" name="" of="" pps="" pre="" ssh="" this="" will="" with="" your=""></admin-]></pulse-config></pre>		PPS Config Data PPS config data				
SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.		<pulse-config><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net<td>n><admin-< td=""><td>username>admin<!--</td--><td>admin-username>·</td><td><admin-j< td=""></admin-j<></td></td></admin-<></td></dns-domain></pulse-config>	n> <admin-< td=""><td>username>admin<!--</td--><td>admin-username>·</td><td><admin-j< td=""></admin-j<></td></td></admin-<>	username>admin </td <td>admin-username>·</td> <td><admin-j< td=""></admin-j<></td>	admin-username>·	<admin-j< td=""></admin-j<>
		SSH Key Name Name of an existing EC2 KeyPair. Your PPS will launch with this KeyPair.				
						•
						_

- Stack name: Specify the stack name in which Pulse Policy Secure needs to be deployed
- Existing VPC ID: Virtual private cloud ID
- Internal Subnet ID: Subnet from which Pulse Policy Secure internal interface needs to lease IP
- External Subnet ID: Subnet from which Pulse Policy Secure external interface needs to lease IP
- PPS AMI ID: ID of the uploaded AMI file
- Instance Type: Size of the instance t2.medium or t2.large
- **PPS Config Data**: Provisioning parameters in an XML format. For details, see Pulse Policy Secure Provisioning Parameters.
- SSH Key Name: This key is used to access PPS via SSH. The SSH keys are generated using sshkeygen on Linux and OS X, or PuTTyGen on Windows. For details about generating the SSH key pairs, refer <u>http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html</u>
- 4. Review the specified details and click **Create stack**. Observe the deployed PPS in a few minutes.

Pulse Policy Secure Provisioning Parameters

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

<pulse-config>

<primary-dns><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>

#	Parameter Name	Туре	Description
1	primary-dns	IP address	Primary DNS for Pulse Policy Secure
2	secondary-dns	IP address	Secondary DNS for Pulse Policy Secure
3	wins-server	IP address	Wins server for Pulse Policy Secure
4	dns-domain	string	DNS domain of Pulse Policy Secure
5	admin-username	string	admin UI user name
6	admin-password	string	admin UI password
7	cert-common-name	string	Common name for the self-signed certificate
8	cert-random-text	string	generation. This certificate is used as the device
9	cert-organization	string	Random text for the self-certificate generation Organization name for the self-signed certificate generation
10	config-download-url	String URL	Http based URL where XML based Pulse Policy

The below table depicts the details of the xml file.

			Secure configuration can be found. During provisioning, Pulse Policy Secure fetches this file and comes up with preloaded configuration. XML based configuration can be present in another VM in AWS cloud or at corporate network which is accessible for Pulse Policy Secure through site to site VPN between AWS and corporate data center
11	config-data	string	base64 encoded XML based Pulse Policy Secure configuration
12	auth-code-license	string	Authentication code that needs to be obtained from Pulse Secure
13	enable-license- server	string	If set to ' y ', PPS will be deployed as a License server. If set to ' n ', PPS will be deployed as a normal server.
14	accept-license- agreement	string	This value is passed to the instance for configuration at the boot time. By default, this value is set to "n". This value must be set to "y" .
15	enable-rest	string	If set to ' y ', REST API access for the administrator user is enabled.

i

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

Configuring Licenses on the Pulse Policy Secure Appliance

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

The Pulse License server can be made available in the corporate network

Pulse License Server in Corporate Network

Figure 20: Pulse License Server in a Corporate Network



Pulse License Server in Cloud Network

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



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

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

Figure 21: Pulse License Server in Cloud Network



Adding Authentication Code in PPS Admin Console

To add Authentication code:

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

Including Authentication Code in CloudFormation Template

To include Authentication code in the CloudFormation template:

- 1. In the CloudFormation template, go to the PPSConfig 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 Policy Secure Virtual Appliance

The Pulse Policy Secure virtual appliance can be accessed:

- <u>as an administrator</u>
- <u>as an end user</u>
- <u>using SSH console</u>

Accessing the Pulse Policy Secure Virtual Appliance as an Administrator

In the AWS portal, navigate to CloudFormation section. Select the stack where PPS is deployed and then click on the 'Outputs' tab. Note down the PPS management, internal and external address from the table as shown in Figure 17.



Services v	Resource Groups 🐱 🔭		🗘 👘 🖓 🗸 N. California 🗸 Support 🗸
CloudFormation \times	CloudFormation > Stacks > NIC3existing		
Stacks	⊡ Stacks (52)	NIC3existing	Delete Update Stack actions V Create stack V
Drifts	Q Filter by stack name	Stack info Events Resources Outputs Parameters Template Change set	ts
StackSets	Active View nested		
Exports	× · /	Outputs (4)	G
Designer	sreebI-AWS-GW1 ○ 2020-03-02 09:39:26 UTC+0530 ⓒ CREATE_COMPLETE	Q Search outputs	©
CloudFormation registry	awsgw335ips	Key 🔺 Value	▼ Description ▼ Export name ▼
Resource types	CREATE_COMPLETE	ExternalAddress Public IP address: 18.144.183.161 Private IP address: 10.200.12.37	PPS Extenal Interface details -
	NIC3existing •	InstanceId i-0dca577d7a6071826	Instance Id of newly created instance -
	2020-02-27 15:23:47 UTC+0530 CREATE_COMPLETE	InternalAddress Public IP address: 18.144.129.40 Private IP address: 10.200.11.87	PPS Internal Interface details -
B	PPS2NIC-1	ManagementAddress Public IP address: 52.8.46.139 Private IP address: 10.200.13.124	PPS Management Interface details -

Use the credentials provided in the provisioning parameters to log in as the administrator https://<PPS-IP>/admin. The default PPS 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 Policy Secure Virtual Appliance administrator can check troubleshooting tools provided in the Pulse Policy Secure admin UI (System->Maintenance->Troubleshooting), to verify whether Pulse Policy Secure is able to reach other cloud resources as well as corporate resources. For this, AWS network administrator needs to ensure that all other resources have Pulse Policy Secure Internal interface as its default gateway.

Accessing the Pulse Policy Secure Virtual Appliance as an End User

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

Figure 23: Pulse External Interface

Services ~	Resource Groups 🗸 🦹		↓ ja v N. California v Support v
CloudFormation \times	CloudFormation > Stacks > NIC3existing		
Stacks Stack details Drifts	C C C C C C C C C C C C C C C C C C C	NIC3existing Stack info Events Resources Outputs Parameters Template Change see	Delete Update Stack actions V Create stack V
StackSets	Active View Hested		
Exports		Outputs (4)	
Designer	sreebI-AWS-GW1 2020-03-02 0939:26 UTC+0530	Q. Search outputs	©
CloudFormation registry	awsgw335ips	Key 🔺 Value	♥ Description ♥ Export name ♥
Resource types	CREATE_COMPLETE	ExternalAddress Public IP address: 18.144.183.161 Private IP address: 10.200.12.37	PPS Extenal Interface details -
	NIC3existing •	InstanceId i-Odca577d7a6071826	Instance Id of newly created instance -
	2020-02-27 15:23:47 UTC+0530 CREATE_COMPLETE	InternalAddress Public IP address: 18.144.129.40 Private IP address: 10.200.11.87	PPS Internal Interface details -
Descione and a	PPS2NIC-1	ManagementAddress Public IP address: 52.8.46.139 Private IP address: 10.200.13.124	PPS Management Interface details -

Accessing the Pulse Policy Secure Virtual Appliance using SSH Console

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

On Linux and Mac OSX

Execute the following command:

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

On Windows

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

tegory.		
Session	Basic options for your PuTTY se	ession
	Specify the destination you want to conne	ect to
	Host Name (or IP address)	Port
Rejudad Bell Features Window Appearance Behaviour Translation Colours Connection Potat Proxy Telnet Rlogin SSH Serial	13.71.121.15	6667
	Connection type: ◎ Raw ◎ Telnet ◎ Rlogin ● SS	H 🔘 Serial
	Load, save or delete a stored session Saved Sessions	
	Default Settings	Load Save Delete
	Close window on exit: Always Never Only on o	clean exit

Figure 24: Putty Configuration – Basic Options

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

Figure 25: Putty Configuration – SSH Authentication



System Operations

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

Figure 26: 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 Policy Secure creates the Pulse Policy 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: <u>https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-ip-addressing.html</u>

IP Addressing Modes

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

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

Modifying Network Parameters After Deployment

Since Networking Infrastructure is provided by AWS, a PPS 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 PPS Virtual Interface.

So, the question is, among eth0, eth1 and eth2 which network interface will become external, internal or management interface? Below table answers this question.

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

Then, question is how you can control the order of network interfaces named eth0, eth1 and eth2 created through CloudFormation template?

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

```
'EC2Instance": {
 "Type": "AWS::EC2::Instance",
 "Properties": {
  "ImageId": {"Ref": "PPSImageAMIId"},
  "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::|oin": [ "-", [ { "Ref": "AWS::StackName" }, "PPSvAWS" ] ] }
   }
   ],
   "UserData": {"Fn::Base64": {"Fn::Join": ["", [{"Ref": "PPSConfigData"}]]}}
}
},
```

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

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

Backing up Configs and Archived Logs on S3 Bucket

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

Configuring Backup Configs and Archived Logs via PPS Admin Console

To configure backing up configs and archived logs:

- 1. Log into the Pulse Policy Secure admin console.
- 2. Navigate to Maintenance > Archiving > Archiving Servers.
- 3. In the Archive Settings section, select the **AWS** option and configure S3 Bucket Name, AWS Access Key, AWS Secret Key, S3 Bucket Location and Destination Path Prefix.

Figure 27: AWS Archive Settings

✓ Archive Settings

Method:	SCP O FTP AWS S3	 Azure Storage
*S3 Bucket Name:	polsecstorageacct	AWS S3 bucket name
*Region:	India	AWS S3 bucket location
*AWS Access Key:	testaws	AWS account access key
*AWS Secret Key:		AWS account secret key
Destination Path Prefix:		Path to copy files under S3 bucket, eg: folder1/folder2
Test Connection		
* indicates required field		

Description Parameter To create an S3 bucket: S3 Bucket Name 1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/. 2. Select Create bucket. 3. 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 S3 bucket location. Region To create AWS Access Key and AWS Secret Key: AWS Access Key 1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/. 2. In the navigation bar on the upper right, select your user name, and then select My Security Credentials. 3. On the AWS IAM Credentials tab, in the Access keys for CLI, SDK, and API access section, select Create access key. 4. 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 <u>https://help.bittitan.com/hc/en-</u> <u>us/articles/115008255268-How-do-I-find-my-AWS-Access-Key-and-Secret-</u>
	<u>Access-Key-</u>
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/ison
"archive-path": "folder1/folder2",
"directory": "ap-south-1",
"method": "AWS",
"Password-cleartext": "xkjdsklukjkwej",
"server": "S3-server-storage-bucket",
"user-name": "ADDDDDFVFFFQXXXXA"
)

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 Policy Secure

To decommission Pulse Policy Secure, perform the following steps:

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

Figure 28: Delete Stack

aws Services 🗸	Resource Groups 🐱 🔸		🗘 sachin 👁 n	wlaaga 👻 N. California 👻 Support 👻
CloudFormation \times	CloudFormation > Stacks			
Stacks	Stacks (52)	0	Delete Update Stack	actions 🔻 Create stack 🔻
Stack details	O Filter by stack name		Activo	View posted
Drifts			Active	View Hested
StackSets				< 1 > 🔘
Exports	Stack name	Status	Created time	Description
Designer	Sreebl-AWS-GW1	⊘ CREATE_COMPLETE	2020-03-02 09:39:26 UTC+0530	Deploy PCS into a new VPC
	O awsgw335ips	⊘ CREATE_COMPLETE	2020-02-27 15:56:52 UTC+0530	Generate internal and external gat
CloudFormation registry	NIC3existing	CREATE_COMPLETE	2020-02-27 15:23:47 UTC+0530	Deploy PPS into an existing VPC

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
PPS VM (t2.medium / t2.xlarge / t2.2xlarge)	Compute	Yes
Virtual Private Cloud with four subnets	Networking	No
Three NICs named PPSInternalNIC, PPSExternalNIC and PPSManagementNIC	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
PPS IMG file of size 40GB in S3 bucket	Storage	Yes
PPS Snapshot file of size 40GB in Elastic block store	Storage	Yes

Limitations

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

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

Troubleshooting

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

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

Figure 29: Boot Diagnostics

aws Services	🗸 🛛 Resource Groups 🗸 🔹		众 sachin @ p	oulseqa 👻 N. California 👻 Support 👻
New EC2 Experience Tell us what you think	Launch Instance Connect	Actions A		⊥ ⊕ ¢ Ø
Instance Types	Q search : existing Add filter	Connect Get Windows Password		(
Launch Templates New	Name • Instance ID	Create Template From Instance	ability Zone 👻 Instance State 👻 Stat	us Checks 👻 Alarm Status Public DNS (IP
Spot Requests	pps-existing i-00e57a4242		est-1a 🥥 running 🥝 2	2/2 checks None 🍡 🍾 ec2-54-153-85-'
Savings Plans	NIC3existing i-0dca577d7a	Instance Settings	Add/Edit Tags	/2 checks None 🍡 🍃 ec2-18-144-129
Reserved Instances	4	Image 🕨 🕨	Attach to Auto Scaling Group	4
Dedicated Hosts	Instance: i-0dca577d7a6071826 (N	Networking	Attach/Replace IAM Role	
Capacity Reservations		ClassicLink	Change Instance Type	
▼ IMAGES AMIS	Description Status Checks	i-0dca577d7a6071826	Change Termination Protection View/Change User Data Change Shutdown Behavior	ec2-18-144-129-40.us-west-
Bundle Tasks	Instance state	running	Change T2/T3 Unlimited Get System Log	1.compute.amazonaws.com 18.144.129.40
ELASTIC BLOCK	Instance type Finding	t2.medium Opt-in to AWS Compute Optimizer for recommendations. Learn more	Get Instance Screenshot Modify Instance Placement	- 18.144.129.40* 18.144.183.161*
Snapshots	Drivete DNO	in 10 200 11 07 up west	Modify Capacity Reservation Settings	52.8.46.139*
Lifecycle Manager	Private DNS	1.compute.internal	Availability zone	us-west-1a
	Private IPs	10.200.11.87, 10.200.13.124,	Security groups	NIC3existing-

The system logs window is displayed.

Figure 30: System Logs

aws Services	System Log: i-0dca577d7a6071826 (NIC3existing-PPSvAWS) C ×
New EC2 Experience Tell us what you think	Press any key to continue.
Instance Types	Press any key to continue. Press any key to continue. Press any key to continue. Press any key to continue.
Launch Templates New	Press any key to continue. Press any key to continue.
Spot Requests Savings Plans	Press any key to continue. Press any key to continue. Press any key to continue.
Reserved Instances	Press any key to continue. Press any key to continue.
Dedicated Hosts	Press any key to continue. Press any key to continue. [HI]]
	GNU GRUB version 0.97 (632K lower / 3931136K upper memory)
AMIs Bundle Tasks	Use the ^ and v keys to select which entry is highlighted. Press enter to boot the selected OS or 'p' to enter a password to unlock the next set of features. [5;78H [m][7m][5;3H Pulse Secure Virt LILO 24.2
ELASTIC BLOCK	Welcome to IVE boot! Current version: 9.1R4 Build 2835 Factory reset version: 9.1R4 Build 2835 boot: Loading current
Snapshots	BIOS data check successful Starting system software version 9.1R4 (build 2835)

Frequently Asked Questions

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 31: 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 32: 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 33: 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.





Pulse Policy Secure, when provisioned through the CloudFormation template provided by Pulse Secure, creates four subnets under a virtual private cloud named "PPSVirtualNetwork". The four Subnets are:

- 1. PPSInternalSubnet
- 2. PPSExternalSubnet
- 3. PPSManagementSubnet

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

- 1. SGExternalSubnet
- 2. SGInternalSubnet
- 3. SGManagementSubnet

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

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

Figure 35: Stack-PPSvExtSG - Inbound Rules

Inbound rules				Edit inbound rules
Туре	Protocol	Port range	Source	Description - optional
НТТР	ТСР	80	0.0.0/0	-
PostgreSQL	ТСР	5432	0.0.0/0	-
Custom TCP	ТСР	11122 - 11123	0.0.0/0	-
Custom TCP	ТСР	49	0.0.0/0	-
Custom UDP	UDP	1812 - 1813	0.0.0/0	-
Custom TCP	ТСР	601	0.0.0/0	-
Custom UDP	UDP	67	0.0.0/0	-
Custom UDP	UDP	162	0.0.0/0	-
Custom UDP	UDP	3799	0.0.0/0	-
HTTPS	ТСР	443	0.0.0/0	-
All ICMP - IPv4	ICMP	All	0.0.0/0	-
Custom UDP	UDP	514	0.0.0/0	-

Figure 36: Stack-PPSvExtSG - Outbound Rules

Inbound rules Outb	ound rules Tags			
Outbound rules				Edite web word webs
Outbound rules				Edit outbound rules
Туре	Protocol	Port range	Destination	Description - optional
All traffic	All	All	127.0.0.1/32	-

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

Figure 37: Stack-PPSvIntSG - Inbound Rules

Inbound rules				Edit inbound rules
Туре	Protocol	Port range	Source	Description - optional
НТТР	ТСР	80	0.0.0/0	
Custom TCP	ТСР	6667	0.0.0/0	-
PostgreSQL	ТСР	5432	0.0.0/0	-
Custom TCP	ТСР	11122 - 11123	0.0.0/0	-
Custom TCP	ТСР	49	0.0.0/0	-
Custom UDP	UDP	1812 - 1813	0.0.0/0	-
Custom TCP	ТСР	601	0.0.0/0	-
Custom UDP	UDP	67	0.0.0/0	
Custom UDP	UDP	162	0.0.0/0	-
Custom UDP	UDP	3799	0.0.0/0	-
HTTPS	ТСР	443	0.0.0/0	
All ICMP - IPv4	ICMP	All	0.0.0/0	-
Custom UDP	UDP	514	0.0.0.0/0	-

Figure 38: Stack-PPSvIntSG - Outbound Rules

Inbound rules Outbo	und rules Tags			
Outbound rules				Edit outbound rules
Туре	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0/0	

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

Figure 39: Stack-PPSvMgmtSG - Inbound Rules

Description	Inbound Rules	Outbound Rules	Tags	
Edit rules				
Туре 🕕	Protocol (j)	Port Range ()	Source (j)	Description (j)
HTTP	TCP	80	0.0.0/0	
Custom TCP Rule	TCP	830	0.0.0/0	
HTTPS	TCP	443	0.0.0/0	
All ICMP - IPv4	All	N/A	0.0.0/0	

Figure 40: Stack-PPSvMgmtSG - Outbound Rules

Pulse Policy Secure Virtual Appliance on Amazon Web Services - Deployment Guide

Description	Inbound Rules	Outbound Rules	Tags	
Edit rules				
Туре ()	Protocol (j)	Port Range (j)	Destination (j)	Description (j)
All traffic	All	All	127.0.0.1/32	

Appendix B: Pulse Policy Secure CloudFormation Template

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

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

The template has following sections:

Parameters	This section defines the parameters used for deploying PPS on 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 PPS on AWS.

Parameters

Key Name: This is the name of the PPS Storage Account where the PPS 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 PPS will launch with this KeyPair.",

"ConstraintDescription": "Must be the name of an existing EC2 KeyPair."
```

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

```
"PPSImageAMIId" : {

"Type" : "String",

"Description" : "AMI ID of your existing PPS image"

},
```

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

], "ConstraintDescription": "Must be an allowed EC2 instance type."

},

PPS 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 Policy Secure through site-to-site VPN between AWS and the corporate data center.

Pulse Policy 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 Policy Secure Provisioning Parameters.

"PPSConfigData" : {

"Type" : "String",

"Description" : "PPS config data",

"Default" : "<pulse-config><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net</dns-domain><admin-username>admin</admin-username><admin-password>password1234</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></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></config-data></c

},



"VPCCIDR": { "Description": "CIDR block for entire VPC.", "Type": "String", "Default": "10.200.0.0/16", "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])\\.}{3}([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])\\.}{3}([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])\\.}{3}([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])\\.}{3}([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])\\.}{3}([0-9]|[1-2][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.}{3}([0-9]|[1-2][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.}{3}([0-9]|[1-2][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.}{3}([0-9]|[1-2][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.}{3}([0-9]|[1-2][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.}{3}([0-9]|[1-2][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.}{3}([0-9]|[1-2][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.}{3}([0-9]|[1-2][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])\\.}{3}([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])\\.}{3}([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])\.}{3}([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])\.}{3}([0-9]|[1-9][0-9]|1[0-9][0-9]|25[0-5])\.}{3}([0-9]|[1-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9][0-9]|1[0-9]|1[0-9][0-9]|1[0-9]|1[0-9][0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-9]|1[0-

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

"InternalSubnetCIDR": {

```
"Description": "PPS internal interface connects to this subnet",

"Type": "String",

"Default": "10.200.11.0/24",

"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])(\V([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 Policy Secure External Interface needs to lease IP.

```
"ExternalSubnetCIDR": {
    "Description": "PPS external interface connects to this subnet",
    "Type": "String",
    "Default": "10.200.12.0/24",
    "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])\\.){3}([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])\\.){3}([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])\\.){3}([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])\\.){3}([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])\\.){3}([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])\\.){3}([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])\\.){3}([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])\\.){3}([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])\\.){3}([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])\\.){3}([0-9]|1[0-9]|2](0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|1[0-9]|2](0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\\.){3}([0-9]|1[0-9]|2](0-9)|2[0-9]|2[0-9]|25[0-5])\
```

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

"ManagementSubnetCIDR": {

```
"Description": "PPS management interface connects to this subnet",

"Type": "String",

"Default": "10.200.13.0/24",

"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])(\V([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 PPS Internal interface.

```
"IntSubnet" : {
```

"Type" : "AWS::EC2::Subnet",

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

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

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

"MgmtSubnet" : {

"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",

PPSvExternalSecurityGroup:

"PPSvExternalSecurityGroup": { "Type": "AWS::EC2::SecurityGroup",

PPSvInternalSecurityGroup:

"PPSvInternalSecurityGroup": { "Type": "AWS::EC2::SecurityGroup",

PPSvManagementSecurityGroup:

"PPSvManagementSecurityGroup": { "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 PPS 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",
"PrimaryPrivatelpAddress"] }]]},
   "Description" : "PPS Management Interface details"
  },
  "ExternalAddress" : {
   "Value" : { "Fn::Join" : [" ", [ "Public IP address:", { "Ref" : "EIP1" }, "Private IP address:", { "Fn::GetAtt" : ["Eth1",
"PrimaryPrivatelpAddress"] }]]},
   "Description" : "PPS Extenal Interface details"
  },
  "InternalAddress" : {
     "Value" : { "Fn::Join" : [" ", [ "Public IP address:", { "Ref" : "EIP0" }, "Private IP address:", { "Fn::GetAtt" : ["Eth0",
"PrimaryPrivatelpAddress"] }]]},
    "Description" : "PPS Internal Interface details"
   }
}
}
```

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

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

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

The template has following sections:

Parameters	This section defines the parameters used for deploying PPS on 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 PPS on AWS.

Parameters

Key Name: This is the name of the PPS Storage Account where the PPS 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 PPS will launch with this KeyPair.",
 "ConstraintDescription": "Must be the name of an existing EC2 KeyPair."

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

```
"PPSImageAMIId" : {

"Type" : "String",

"Description" : "AMI ID of your existing PPS image"

},
```

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

], "ConstraintDescription": "Must be an allowed EC2 instance type."

},

PPS 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 Policy Secure through site-to-site VPN between AWS and the corporate data center.

Pulse Policy 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 Policy Secure Provisioning Parameters.

"PPSConfigData" : {

"Type" : "String",

"Description" : "PPS config data",

"Default" : "<pulse-config><wins-server>1.1.1.1</wins-server><dns-domain>psecure.net</dns-domain><admin-username>admin</admin-username><admin-password>password1234</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></config-data></config-data><auth-code-license></auth-code-license><enable-license-server>n</enable-license-server><accept-license-agreement></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 PPS Internal interface connects.

```
"SubnetIntId" : {

"Type" : "String",

"Description" : "ID of the subnet where PPS internal interface connects"

},
```

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

```
"SubnetExtId" : {

"Type" : "String",

"Description" : "ID of the subnet where PPS External interface connects"

},
```

SubnetMgmtld: This is the ID of the subnet to which PPS Management interface connects.

```
"SubnetMgmtId" : {
    "Type" : "String",
    "Description" : "ID of the subnet where PPS 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",

PPSvExternalSecurityGroup:

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

PPSvInternalSecurityGroup:

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

PPSvManagementSecurityGroup:

"PPSvManagementSecurityGroup": { "Type": "AWS::EC2::SecurityGroup",

EC2Instance:

"EC2Instance" : {

"Type" : "AWS::EC2::Instance", "DependsOn" : ["EIPAssoc0", "EIPAssoc1", "EIPAssoc2"],

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 PPS 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",
"PrimaryPrivatelpAddress"] }]]},
   "Description" : "PPS Management Interface details"
  },
  "ExternalAddress" : {
   "Value" : { "Fn::Join" : [" ", [ "Public IP address:", { "Ref" : "EIP1" }, "Private IP address:", { "Fn::GetAtt" : ["Eth1",
"PrimaryPrivatelpAddress"] }]]},
   "Description" : "PPS Extenal Interface details"
  },
  "InternalAddress" : {
     "Value" : { "Fn::Join" : [" ", [ "Public IP address:", { "Ref" : "EIPO" }, "Private IP address:", { "Fn::GetAtt" : ["Eth0",
"PrimaryPrivatelpAddress"] }]]},
     "Description" : "PPS 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.