

VYATTA, INC.



Vyatta System

# Amazon Machine Image

INSTALLATION AND OPERATIONS GUIDE



Vyatta  
Suite 200  
1301 Shoreway Road  
Belmont, CA 94002  
[vyatta.com](http://vyatta.com)  
650 413 7200  
1 888 VYATTA 1 (US and Canada)

## **COPYRIGHT**

Copyright © 2005–2011 Vyatta, Inc. All rights reserved.

Vyatta reserves the right to make changes to software, hardware, and documentation without notice. For the most recent version of documentation, visit the Vyatta web site at [vyatta.com](http://vyatta.com).

## **PROPRIETARY NOTICES**

Vyatta is a registered trademark of Vyatta, Inc.

VMware, VMware ESX, and VMware server are trademarks of VMware, Inc.

XenServer, and XenCenter are trademarks of Citrix Systems, Inc.

All other trademarks are the property of their respective owners.

RELEASE DATE: September 2011

DOCUMENT REVISION: R6.3.0 v03

RELEASED WITH: R6.3.0

PART NO. A0-0243-10-0000

# Contents

<b>Preface</b> .....	<b>v</b>
Intended Audience .....	vi
Organization of This Guide .....	vi
Document Conventions .....	vii
Vyatta Publications .....	viii
 <b>Chapter 1 Installing the System</b> .....	 <b>1</b>
Introduction .....	2
Before You Begin .....	2
AWS Reference Material .....	3
Obtaining the Vyatta AMI .....	4
Launching a Vyatta AMI Instance into a VPC .....	5
Create a VPC .....	5
Launch a Vyatta AMI Instance into the VPC .....	5
Modify the Default Security Group .....	10
Assign an AWS Elastic IP Address to the Instance .....	13
Access the Instance Remotely .....	15
Terminating an Instance .....	16
 <b>Chapter 2 Configuration Examples</b> .....	 <b>17</b>
Creating a NAT Device .....	18
Configure the Vyatta AMI Instance for NAT .....	19
Modify the Default Security Group .....	20
Allow the Instance to Be Used for NAT .....	23
Create a Private Subnet .....	23
Create a Route Table for the Private Subnet .....	25
Launch an Instance into the Private Subnet .....	29
Access the Private Instance Remotely .....	34
Verify the Instance is Working as Expected .....	34
Creating a Site-to-site IPsec VPN Connection .....	35
Creating a Site-to-site OpenVPN Connection .....	39
Creating a Remote Access VPN Connection .....	41
 <b>Chapter 3 Upgrading the System</b> .....	 <b>44</b>
Release-Specific Upgrade Information .....	45
Upgrading in a Virtual Environment .....	45

---

**Glossary ..... 47**

# Preface

On the Vyatta Subscription Edition, the Vyatta system is available as an Amazon Machine Image (AMI) for use with Amazon Web Services (AWS).



*This feature is available only in the Vyatta Subscription Edition.*

This document explains how to launch the Vyatta AMI into a VPC within the AWS cloud and to configure AWS such that you can access the Vyatta system remotely. It also provides examples of how to configure the Vyatta system for various uses, and how to upgrade a Vyatta AMI system.

This preface provides information about using this guide. The following topics are presented:

- [Intended Audience](#)
- [Organization of This Guide](#)
- [Document Conventions](#)
- [Vyatta Publications](#)

---

## Intended Audience

---

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security
- IP services

---

## Organization of This Guide

---

This guide has the following chapters:

Chapter	Description	Page
<a href="#">Chapter 1: Installing the System</a>	This chapter describes the Vyatta AMI and how to install it within the Amazon Web Services cloud.	<a href="#">1</a>
<a href="#">Chapter 2: Configuration Examples</a>	This chapter presents examples for configuring a Vyatta AMI instance for a variety of scenarios.	<a href="#">17</a>
<a href="#">Chapter 3: Upgrading the System</a>	This chapter explains how to upgrade Vyatta system software on a Vyatta AMI.	<a href="#">44</a>
<a href="#">Glossary</a>		<a href="#">47</a>

# Document Conventions

This guide uses the following advisory paragraphs, as follows.

**WARNING** Warnings alert you to situations that may pose a threat to personal safety.



**CAUTION** Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service.



**NOTE** Notes provide information you might need to avoid problems or configuration errors.

This document uses the following typographic conventions.

Monospace	Examples, command-line output, and representations of configuration nodes.
<b>bold Monospace</b>	Your input: something you type at a command line.
<b>bold</b>	Commands, keywords, and file names, when mentioned inline.  Objects in the user interface, such as tabs, buttons, screens, and panes.
<i>italics</i>	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs (“+”), as in <Ctrl>+c.
[ key1   key2]	Enumerated options for completing a syntax. An example is [enable   disable].
num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.
arg1..argN	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
arg[ arg...] arg[,arg...]	A value that can optionally represent a list of elements (a space-separated list and a comma-separated list, respectively).

# Vyatta Publications

---

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding the information you need.

Additional information is available on [www.vyatta.com](http://www.vyatta.com) and [www.vyatta.org](http://www.vyatta.org).



# Chapter 1: Installing the System

This chapter describes the Vyatta AMI and how to install it within the Amazon Web Services cloud.

This chapter presents the following topics:

- [Introduction](#)
- [Before You Begin](#)
- [AWS Reference Material](#)
- [Obtaining the Vyatta AMI](#)
- [Launching a Vyatta AMI Instance into a VPC](#)
- [Terminating an Instance](#)

# Introduction

---

Amazon Web Services (AWS) is Amazon's cloud computing service. AWS provides the tools and infrastructure required by businesses to run computing environments "within the cloud."

When you operate a computing environment within the cloud, you reduce capital expenditures to a minimum, and you gain the ability to easily scale up or down your compute resources as required. You pay as you go and you pay only for the resources you use.

AWS provides a number of different products and services to enable businesses to build the environments they require. At the core of AWS is the Amazon Machine Image (AMI). An AMI is a virtual machine template. You instantiate the template as virtual machine instances within the AWS cloud. A variety of AMIs are available from a number of vendors. The Vyatta AMI is a version of the Vyatta Subscription Edition system packaged to run in the AWS cloud.

The Amazon Elastic Compute Cloud (EC2) is the AWS infrastructure within which all AMIs are launched. EC2 allows you to easily obtain and scale compute capacity as required.

A Virtual Private Cloud (VPC) allows you to provision a virtual private network within the AWS cloud. A VPC allows you to define a virtual network topology within which you can create subnets, select IP addresses, and configure routing tables and network gateways.

This document explains how to launch the Vyatta AMI into a VPC within the AWS cloud and to configure AWS such that you can access the Vyatta system remotely. It also provides examples of how to configure the Vyatta system to act as a NAT gateway, a site-to-site IPsec VPN endpoint, a site-to-site OpenVPN endpoint, and a remote access IPsec VPN server.

## Before You Begin

---

To use this guide, and to deploy the Vyatta system within the AWS environment, you must be conversant with AWS and virtual private clouds (VPCs). This guide assumes you are thoroughly familiar with at least the following AWS documentation:

- <http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/>
- <http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide/>

You should also be conversant with the AWS services you will be using. You can find AWS documentation at <http://aws.amazon.com/documentation/>.

This document also assumes the following:

- You have an AWS account. (Sign up for an AWS account at <http://aws.amazon.com/>.)
- You are able to log on to the AWS Management Console.
- You have purchased the Vyatta AMI.
- You have provided Vyatta with your AWS account number so that Vyatta can provide you with access to the Vyatta AMI.
- You have mastered the following AWS skills:
  - Creating a VPC subnet
  - Creating and attaching an Amazon VPC Internet Gateway to the VPC
  - Setting up routing in the VPC to enable traffic to flow between the VPC subnet and the Internet
  - Setting up a security group to control inbound and outbound traffic for the instances launched within the VPC
  - Launching an AMI instance (either Linux/UNIX or Windows) into the VPC
  - Creating a key pair and assigning it to an instance
  - Assigning an Elastic IP address to an instance
  - Connecting to an instance remotely using SSH (for Linux/UNIX instances) or RDP (for Windows instances)

---

**NOTE** You can find your account number using the AWS Management Console. Click **Account** at the top of the the AWS Management Console. Click **Account Activity** on the **Your Account** page. Your account number appears in the top right corner of the **Account Activity** page.

---

## AWS Reference Material

---

For an overview of AWS, see the **Introduction to AWS** webinar in the **Solutions** playlist at <http://aws.amazon.com/resources/webinars/>.

The AWS Documentation library can be found at <http://aws.amazon.com/documentation/>.

Amazon EC2 Documentation index can be found at <http://aws.amazon.com/documentation/ec2/>

The **Amazon EC2 Getting Started Guide** can be found at <http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/>

The **Amazon EC2 User Guide** can be found at <http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/>

Amazon VPC Documentation index can be found at  
<http://aws.amazon.com/documentation/vpc/>

The **Amazon VPC Getting Started Guide** can be found at  
<http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide/>

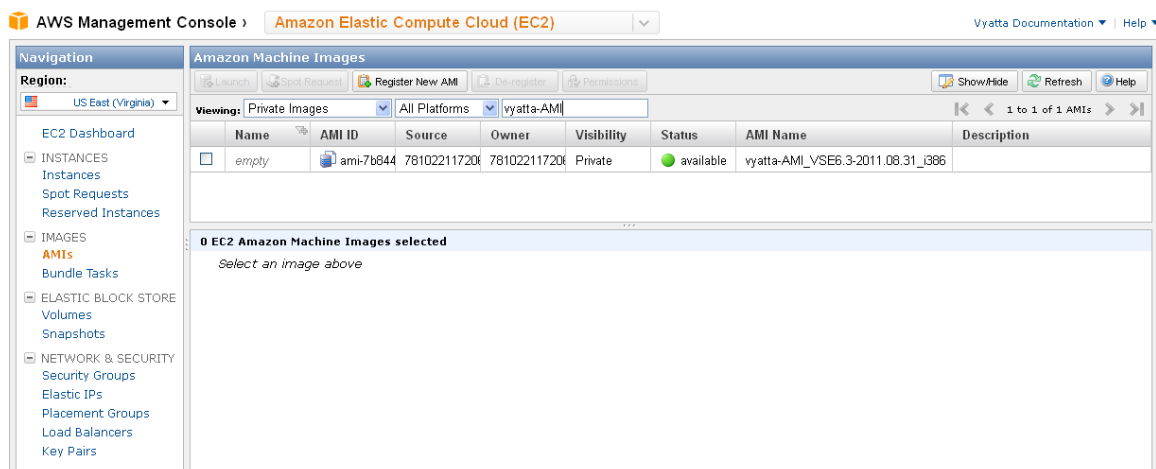
The **Amazon VPC User Guide** can be found at  
<http://docs.amazonwebservices.com/AmazonVPC/latest/UserGuide/>

## Obtaining the Vyatta AMI

When you purchase the Vyatta AMI and provide Vyatta your AWS account number, Vyatta shares the Vyatta AMI with you within the AWS environment.

To view the Vyatta AMI:

- 1 Select **Amazon Elastic Compute Cloud (EC2)** on the dropdown at the top of the AWS Management Console.
- 2 Select **AMIs** in the left navigation pane.
- 3 Within the **Viewing:** field, select **Private Images, All Platforms**.
- 4 Specify **vyatta-AMI** in the search field. All available Vyatta AMIs appear, named **vyatta-AMI\_release number**, where *release number* is the release number of the image, as in the following example.



The Vyatta AMI comes preconfigured as a standard Vyatta Subscription Edition system with some additional configuration changes to ease installation and access within AWS:

- The eth0 interface is configured to use DHCP. The IP address can be specified when launching the instance. If not specified, AWS assigns one automatically. The IP address is in the range of private addresses for the subnet into which it is launched.
- SSH access is configured.
- The host-name is set to **VyattaAMI**.

The Vyatta AMI must be launched within a VPC. It is supported as a **Small** instance type (**m1.small**, **1.7 GB**) within AWS and is provided with persistent Amazon Elastic Block Storage (EBS).

## Launching a Vyatta AMI Instance into a VPC

---

This section presents the following topics:

- [Create a VPC](#)
- [Launch a Vyatta AMI Instance into the VPC](#)
- [Modify the Default Security Group](#)
- [Assign an AWS Elastic IP Address to the Instance](#)
- [Access the Instance Remotely](#)

### Create a VPC

You can create a VPC with a single public subnet by following the steps outlined in the **Amazon VPC Getting Started Guide** at <http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide/>.

The example that follows assumes that you have completed the steps in the **Amazon VPC Getting Started Guide**. These steps create a VPC that provides for addresses in the range of 10.0.0.0/16 and a public subnet in the range of 10.0.0.0/24. The example uses these addresses, but any private IP address ranges defined in RFC 1918 (that is, 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16) can be used.

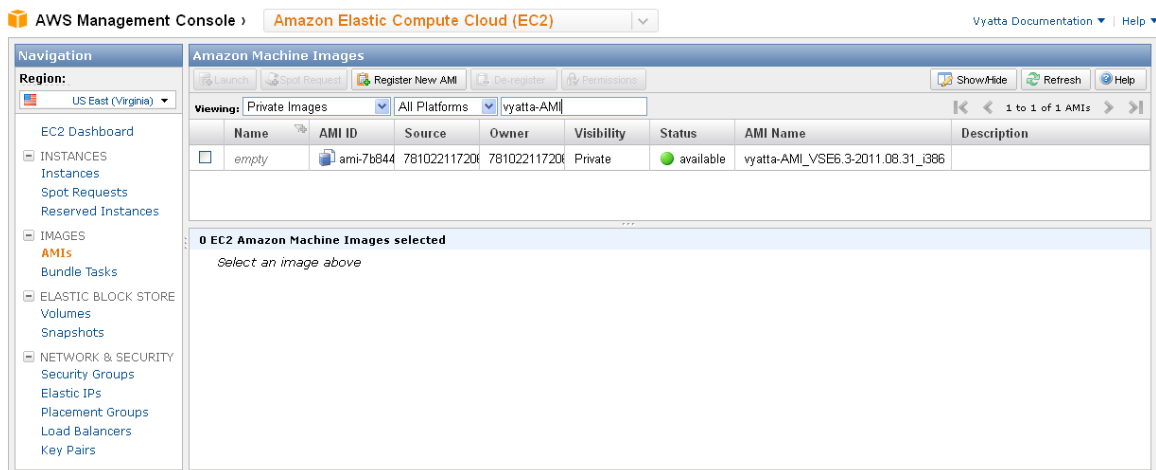
### Launch a Vyatta AMI Instance into the VPC

Launching a Vyatta AMI instance is the same as launching an instance of any other AMI within the AWS cloud.

**To launch a Vyatta AMI instance into a VPC**

- 1 Select **Amazon Elastic Compute Cloud (EC2)** on the dropdown at the top of the AWS Management Console.

- 2 Select **AMIs** in the left navigation pane. The **Amazon Machine Images** page opens on the right.
- 3 In the **Viewing:** field, select **Private Images**, **All Platforms**, and specify **vyatta-AMI** as the search string. Vyatta AMIs are listed.



- 4 Select a Vyatta AMI and click **Launch** at the top of the **Amazon Machine Images** page. The **Request Instances Wizard** opens at the **Instance Details** step.

- Select **Small (m1.small, 1.7GB)** as the **Instance Type**.

**NOTE** If you select **Micro (t1.micro, 613 MB)**, you will not be able to launch the instance into your VPC.

- 5 Select **Launch Instances Into Your Virtual Private Cloud**.
- 6 In the **Subnet ID:** field, select the subnet within the VPC to which to attach the instance (10.0.0.0/24) and click **Continue**. The **Advanced Instance Options** page opens.

**Request Instances Wizard** Cancel

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

Number of Instances: 1

Availability Zone: No Preference

**Advanced Instance Options**

Here you can choose a specific **kernel** or **RAM disk** to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default RAM Disk ID: Use Default

Monitoring: ☐ Enable CloudWatch detailed monitoring for this instance (additional charges will apply)

User Data:

☒ as text ☐ as file ☐ base64 encoded

IP Address: 10.0.0.10  
Optionally specify the IP address of your instance within the 10.0.0.0/24 subnet.

Termination Protection: ☐ Prevention against accidental termination.

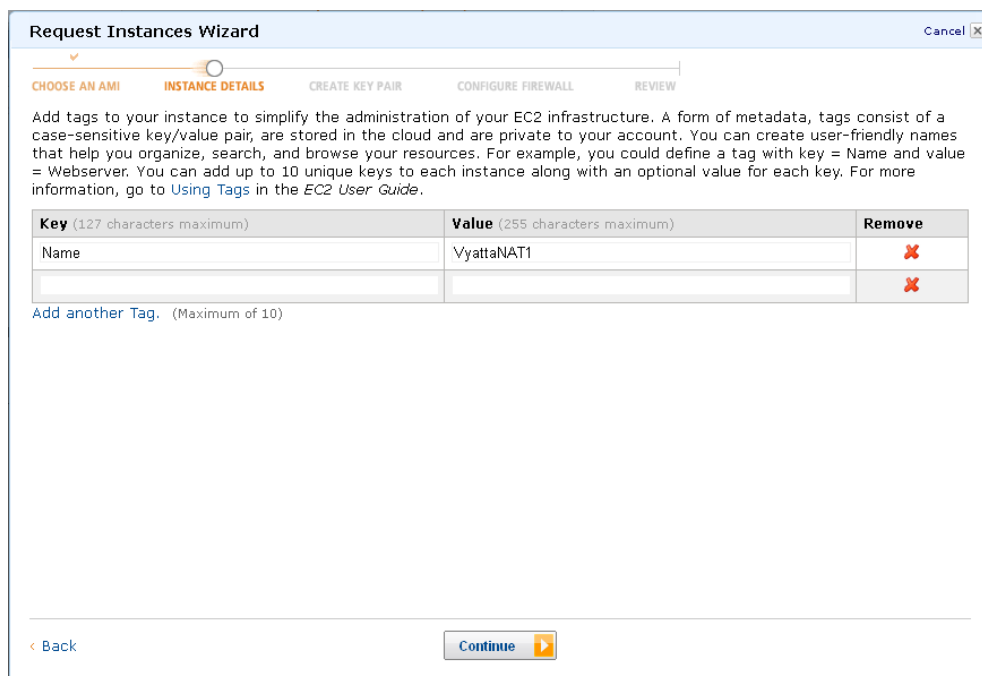
Shutdown Behavior: Stop Choose the behavior when the instance is shutdown from within the instance.

Tenancy: Default

Back Continue

- 7 In the **IP Address:** field, enter 10.0.0.10 and press **Continue**.

Whatever addressing scheme you choose to implement, this address must be within the address range of the public subnet you created—in the example, within the 10.0.0.0/24 address range. The **Add Tags** page appears.



**Request Instances Wizard** Cancel

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

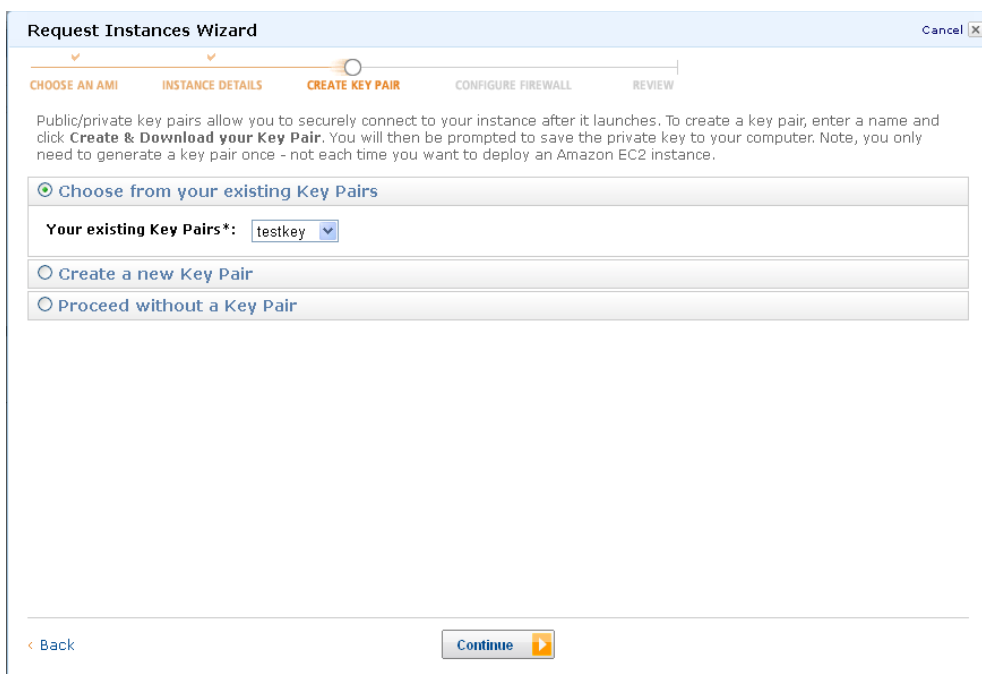
Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. For example, you could define a tag with key = Name and value = Webserver. You can add up to 10 unique keys to each instance along with an optional value for each key. For more information, go to [Using Tags](#) in the *EC2 User Guide*.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
Name	VyattaNAT1	

[Add another Tag.](#) (Maximum of 10)

[< Back](#) [Continue](#)

- 8 In the **Key** column, add a key called **Name**. In the **Value** column, enter **VyattaNAT1** and press **Continue**. The **Create Key Pair** page opens.



**Request Instances Wizard** Cancel

CHOOSE AN AMI **INSTANCE DETAILS** **CREATE KEY PAIR** CONFIGURE FIREWALL REVIEW

Public/private key pairs allow you to securely connect to your instance after it launches. To create a key pair, enter a name and click **Create & Download your Key Pair**. You will then be prompted to save the private key to your computer. Note, you only need to generate a key pair once - not each time you want to deploy an Amazon EC2 instance.

☒ Choose from your existing Key Pairs

Your existing Key Pairs\*:

☐ Create a new Key Pair

☐ Proceed without a Key Pair

[< Back](#) [Continue](#)



- 9 Select **Choose from your existing Key Pairs** and select an existing key pair from the **Your existing Key Pairs** drop-down list. Click **Continue**. The **Configure Firewall** page opens.

**Request Instances Wizard** Cancel

CHOOSE AN AMI INSTANCE DETAILS CREATE KEY PAIR **CONFIGURE FIREWALL** REVIEW

Security groups determine whether a network port is open or blocked on your instances. You may use an existing security group, or we can help you create a new security group to allow access to your instances using the suggested ports below. Add additional ports now or update your security group anytime using the Security Groups page.

☒ Choose one or more of your existing Security Groups

sg-bc100dd0 - default

(Selected groups: sg-bc100dd0)

☐ Create a new Security Group

[< Back](#) [Continue](#)

- 10 Select the default security group and click **Continue**. The **Review** page opens.

**Request Instances Wizard** Cancel

CHOOSE AN AMI INSTANCE DETAILS CREATE KEY PAIR CONFIGURE FIREWALL **REVIEW**

Please review the information below, then click **Launch**.

**AMI:** Other Linux AMI ID ami-7b844712 (i386) [Edit AMI](#)

**Number of Instances:** 1

**VPC ID:** vpc-8388b9ea

**VPC Subnet:** subnet-9888b9f1 (10.0.0.0/24)

**Availability Zone:** No Preference

**Instance Type:** Small (m1.small)

**Instance Class:** On Demand [Edit Instance Details](#)

**Monitoring:** Disabled **Termination Protection:** Disabled

**Tenancy:** Default

**Kernel ID:** Use Default **Shutdown Behavior:** Stop

**RAM Disk ID:** Use Default

**IP Address:** 10.0.0.10

**User Data:** [Edit Advanced Details](#)

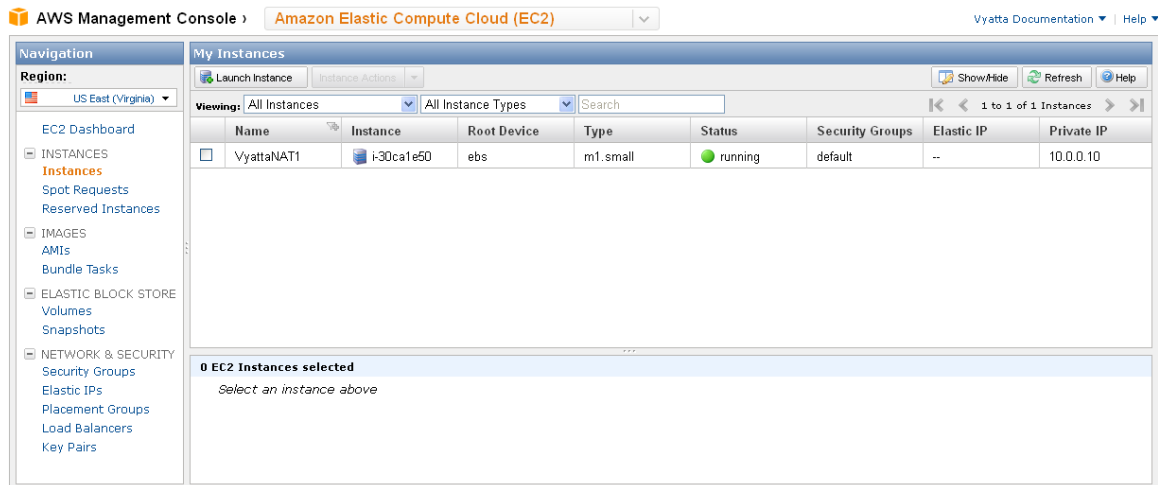
**Key Pair Name:** testkey [Edit Key Pair](#)

**Security Group(s):** sg-bc100dd0 [Edit Firewall](#)

[< Back](#) [Launch](#)

- 11 Review the details for the instance you are creating. When you are satisfied, click **Launch**. The instance starts. Click **Close** on the final wizard screen.

To view the status of the newly launched instance, select **Instances** on the left navigation pane within **Amazon Elastic Compute Cloud (EC2)**.



## Modify the Default Security Group

Security groups provide the policies that control traffic flow and access for EC2 instances and instances within a VPC. EC2 security groups and VPC security groups are independent of one another. EC2 security groups cannot be used for instances within a VPC, and VPC security groups cannot be used for EC2 instances (that is, instances not associated with a VPC). Vyatta AMI instances are launched into VPCs so they use VPC security groups.

The default VPC security group allows instances within the VPC to communicate with one another and to access the Internet, but it does not allow remote access to the instances. To provide remote SSH access into the VPC, either create a new security group and associate the instance with the new group, or modify the default security group. This example modifies the default security group to allow SSH access from anywhere.

---

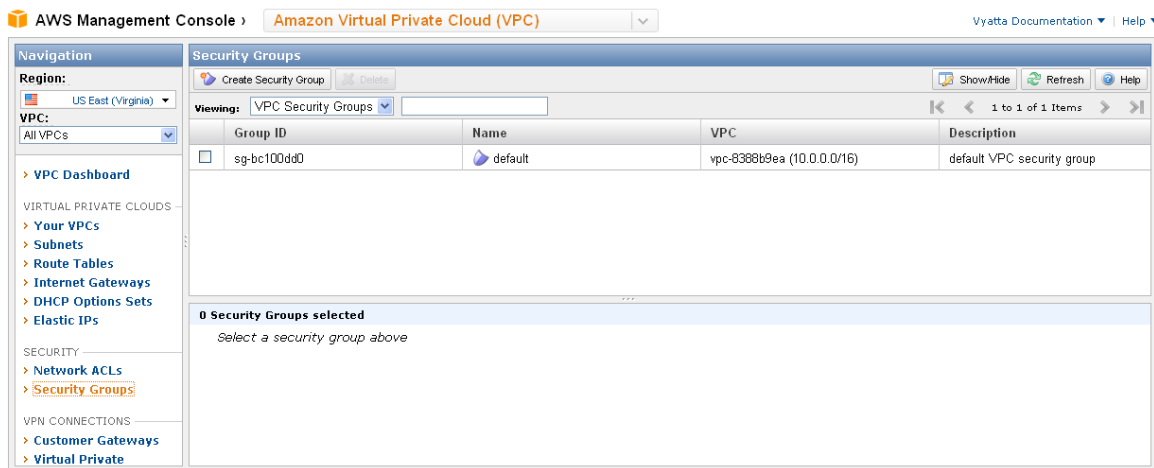
**NOTE** This example allows SSH access from anywhere for testing purposes only. In general, it is best to restrict SSH access to source addresses that you control. Change the port to something other than 22 or 2222. Also, make sure you change the default password on all devices in your network.

---

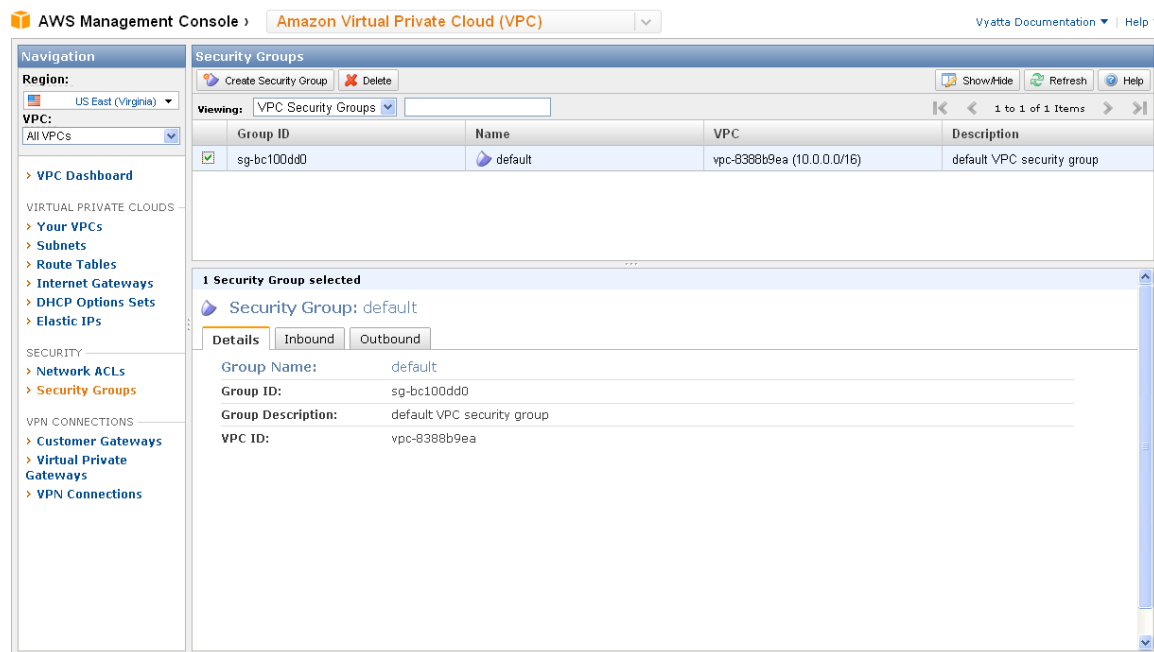
### To modify the default security group to allow SSH access

- 1 Select **Amazon Virtual Private Cloud (VPC)** on the dropdown at the top of the AWS Management Console.

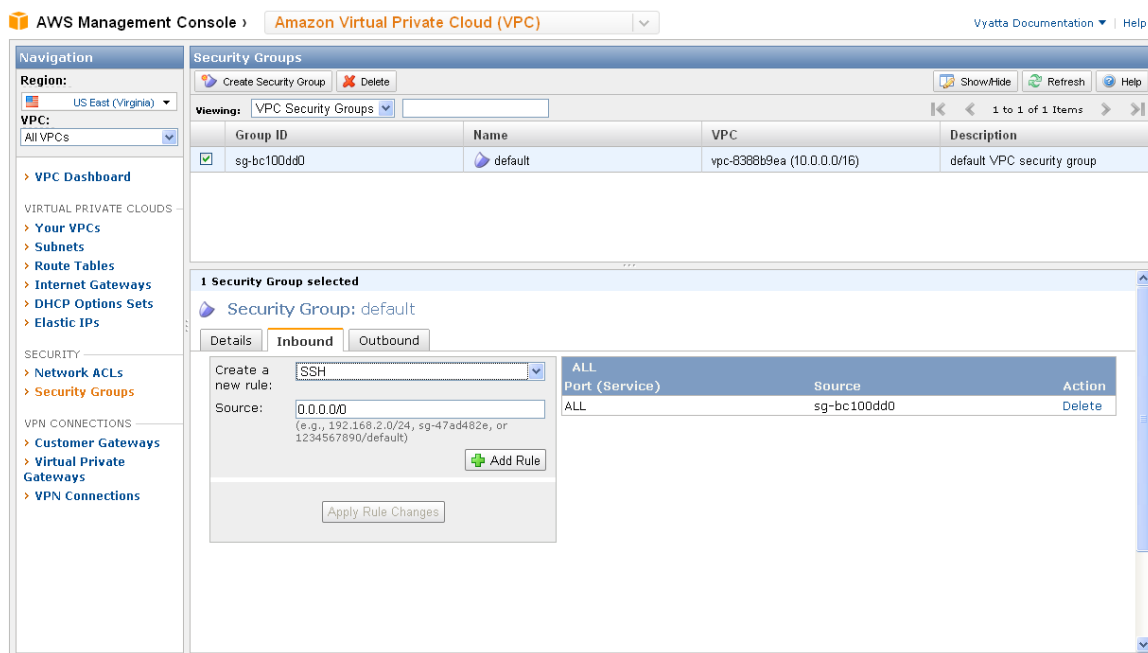
- In the left navigation pane, select **Security Groups**. The **Security Groups** page opens on the right.



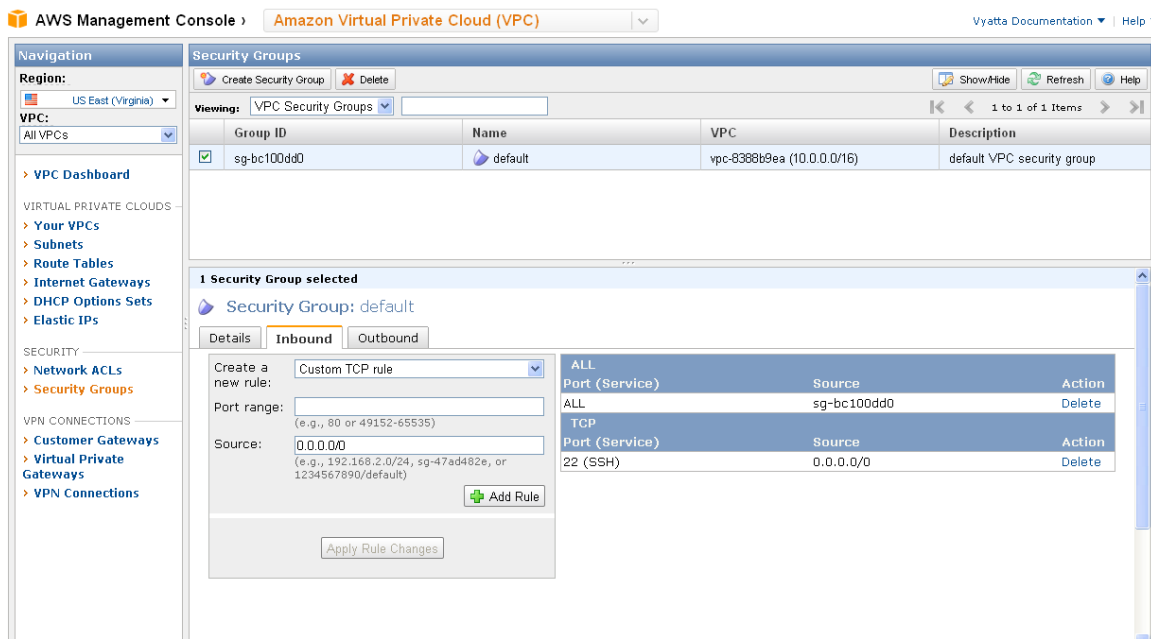
- Select the **default** security group. The details for the default security group appear at the bottom of the page.



- Select the **Inbound** tab. The default inbound rule appears. This rule provides access between the instances that use this security group.



- 5 In the **Create a new rule:** field, select **SSH** from the drop-down menu.
- 6 In the **Source:** field, enter **0.0.0.0/0** and click **Add Rule**. The rule appears in the rule table to the right. Click **Apply Rule Changes** to apply the rule change. The security group now allows SSH access from anywhere.



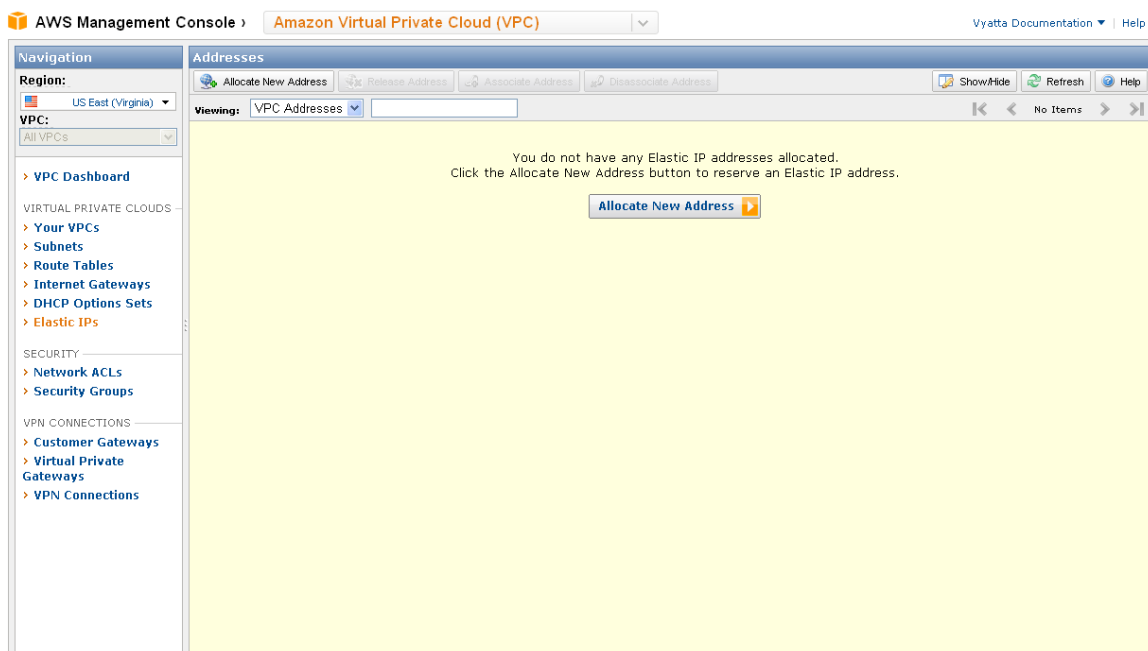
To enable access to the Vyatta Web GUI, you must add a rule that allows HTTPS (port 443) access. You must also configure the HTTPS service on the Vyatta system using the `set service https` command in configuration mode.

## Assign an AWS Elastic IP Address to the Instance

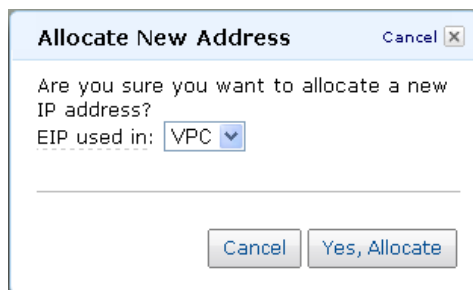
In order to access the instance remotely you assign it an AWS Elastic IP address.

To assign an Elastic IP address

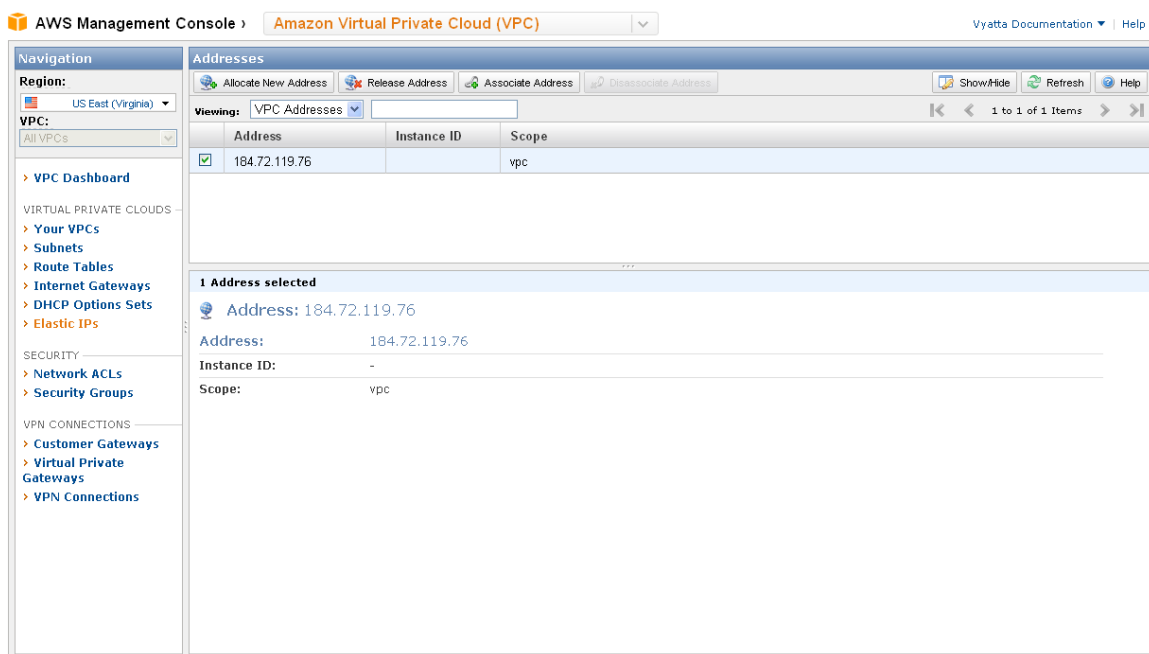
- 1 Select **Amazon Virtual Private Cloud (VPC)** on the dropdown at the top of the AWS Management Console.
- 2 In the left navigation pane, select **Elastic IPs**. The **Addresses** page opens.



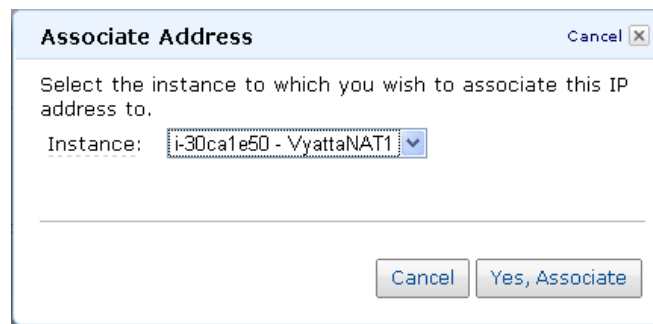
- 3 If you don't already have an Elastic IP address available, click **Allocate New Address**. The **Allocate New Address** dialog opens.



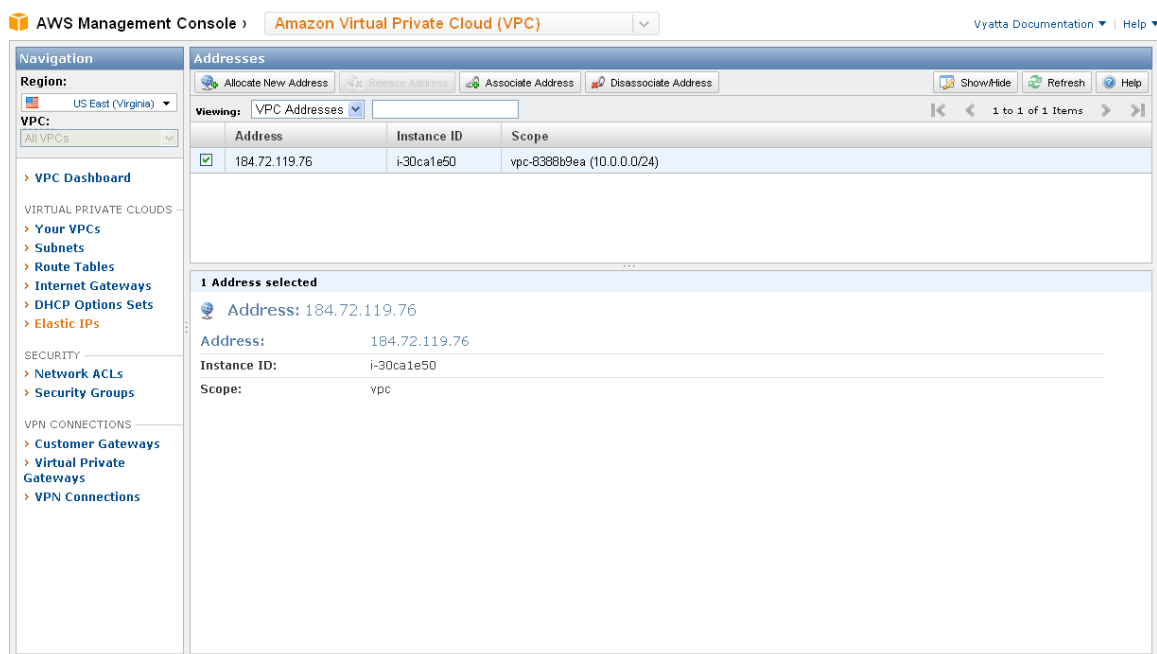
- 4 In the **EIP used in:** field, select **VPC**. Click **Yes, Allocate**. A new Elastic IP address appears on the **Addresses** page.



- 5 Select the Elastic IP address to be associated with the instance you launched. Click **Associate Address**. The **Associate Address** dialog opens.



- 6 In the **Instance:** field, select the instance that you launched above. Click **Yes, Associate**. The Elastic IP address is associated with the instance that you created. This association appears on the **Addresses** page.



## Access the Instance Remotely

After you have modified the security group associated with the instance to allow access from SSH and you have provided the instance with an Elastic IP address, you can test your access to it.

- 1 On a remote machine, open an SSH session. As the destination address, provide the Elastic IP address you associated with the instance.

---

**NOTE** On Linux/UNIX systems use the **ssh** command. On Windows machines use a program such as **putty** for SSH access.

---

- 2 Once connected you will see the **login as:** prompt. To use the default login credentials, log on as user **vyatta** with password **vyatta**.

## Terminating an Instance

---

If you terminate a Vyatta instance, make sure you also remove the storage volume attached to the instance (unless you wish to reuse it). Unless you explicitly delete the storage volume, you are charged for it.



## Chapter 2: Configuration Examples

This chapter presents examples for configuring a Vyatta AMI instance for a variety of scenarios.

This chapter presents the following topics:

- [Creating a NAT Device](#)
- [Creating a Site-to-site IPsec VPN Connection](#)
- [Creating a Site-to-site OpenVPN Connection](#)
- [Creating a Remote Access VPN Connection](#)

## Creating a NAT Device

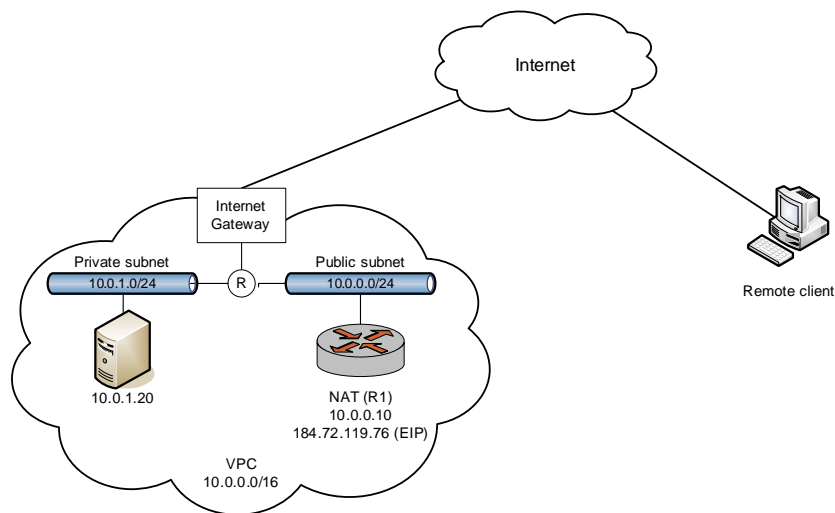
At the end of the installation procedure described in [Chapter 1: Installing the System](#), the following prerequisites for the examples in this chapter were completed:

- A Vyatta AMI instance was launched into an existing VPC with a single public subnet.
- The default security group was modified to allow SSH access.
- An Elastic IP address was assigned to the instance's sole interface.
- Remote SSH access was tested.

In this example, the following steps are completed:

- The Vyatta AMI instance is configured as a Network Address Translation (NAT) device.
- A new subnet is created within the VPC.
- A routing table is configured so that the subnet can route traffic through the Vyatta NAT device.
- A new instance is launched within the new subnet.
- Remote access to the instance in the new subnet is tested using SSH.

The following diagram shows the configuration that is created.



## Configure the Vyatta AMI Instance for NAT

To configure the Vyatta AMI instance to act as a NAT device

- 1 Using SSH and the Vyatta AMI instance's Elastic IP address, log on to the Vyatta AMI instance .

- 2 Enter configuration mode.

```
vyatta@vyatta:~$ configure
[edit]
vyatta@vyatta#
```

- 3 Change the hostname to **R1** to identify the instance.

```
vyatta@vyatta# set system host-name R1
[edit]
vyatta@vyatta#
```

The command prompt changes to reflect the new host name the next time you log in.

- 4 Configure masquerade NAT for outbound traffic from subnet **10.0.1.0/24**. (This network address represents the private subnet to be created in a later step.)

```
vyatta@vyatta# set service nat rule 10 outbound-interface eth0
[edit]
vyatta@vyatta# set service nat rule 10 source address 10.0.1.0/24
[edit]
vyatta@vyatta# set service nat rule 10 type masquerade
[edit]
vyatta@vyatta#
```

- 5 Configure destination NAT to provide remote access to an instance in the private subnet. The NAT rule will pass connections to port 3333 to address 10.0.1.20 port 22. (This instance will be launched in a later step.)

```
vyatta@vyatta# set service nat rule 20 destination port 3333
[edit]
vyatta@vyatta# set service nat rule 20 inbound-interface eth0
[edit]
vyatta@vyatta# set service nat rule 20 inside-address address 10.0.1.20
[edit]
vyatta@vyatta# set service nat rule 20 inside-address port 22
[edit]
vyatta@vyatta# set service nat rule 20 protocol tcp
[edit]
vyatta@vyatta# set service nat rule 20 type destination
[edit]
vyatta@vyatta#
```

- 6 Commit and save the changes.

```
vyatta@vyatta# commit
```

```
[edit]
vyatta@vyatta# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyatta@vyatta#
```

7 View the NAT-related changes.

```
vyatta@vyatta# show service nat
rule 10 {
    outbound-interface eth0
    source {
        address 10.0.1.0/24
    }
    type masquerade
}
rule 20 {
    destination {
        port 3333
    }
    inbound-interface eth0
    inside-address {
        address 10.0.1.20
        port 22
    }
    protocol tcp
    type destination
}
[edit]
vyatta@vyatta#
```

8 Exit configuration mode and then exit the login session.

```
vyatta@vyatta# exit
exit
vyatta@vyatta:~$ exit
```

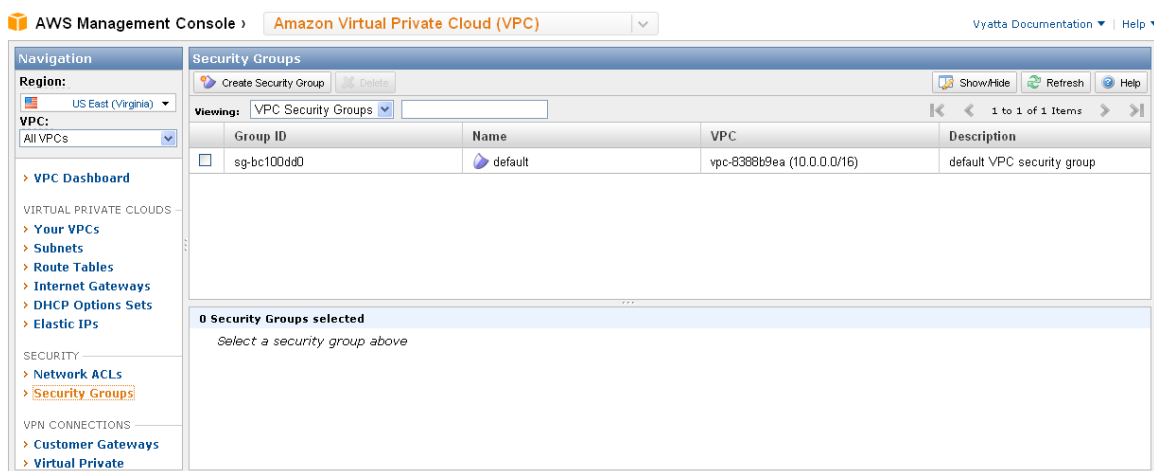
The SSH session terminates.

## Modify the Default Security Group

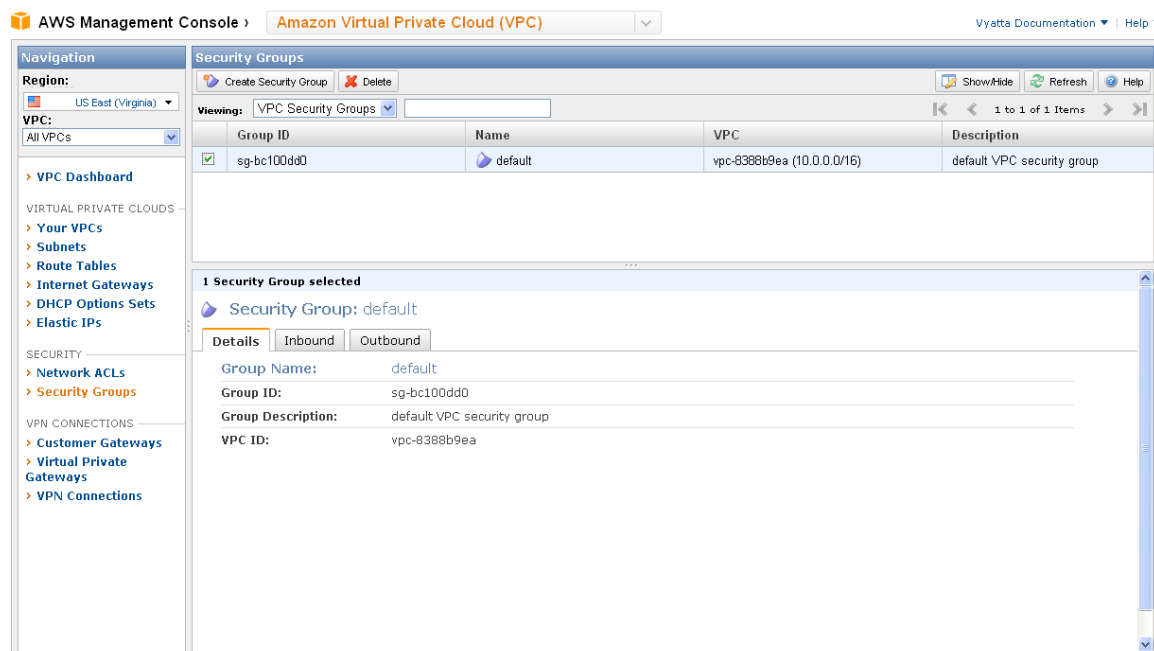
This example modifies the default security group to allow port 3333 access from anywhere. Connections to the Elastic IP address on port 3333 are translated by the Vyatta NAT device and then routed to the private instance that will be created in a later step.

To modify the default security group to allow port 3333 access

- 1 Select **Amazon Virtual Private Cloud (VPC)** on the dropdown at the top of the AWS Management Console.
- 2 In the left navigation pane, select **Security Groups**. The Security Groups page opens on the right.



- 3 Select the **default** security group. The details for the default security group appear at the bottom of the page.



- 4 Select the **Inbound** tab. The current inbound rules appear.

**AWS Management Console** > **Amazon Virtual Private Cloud (VPC)**

Region: US East (Virginia) | VPC: All VPCs

**Security Groups**

Viewing: VPC Security Groups

Group ID	Name	VPC	Description
sg-bc100dd0	default	vpc-8388b9ea (10.0.0.0/16)	default VPC security group

**1 Security Group selected**

Security Group: default

Details | **Inbound** | Outbound

Create a new rule: Custom TCP rule

Port range:

Source:

Port (Service)	Source	Action
ALL	sg-bc100dd0	Delete
TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete

- 5 In the **Create a new rule:** field, select **Custom TCP rule** from the drop-down list.
- 6 In the **Port Range:** field, enter 3333. In the **Source:** field, enter 0.0.0.0/0 and click **Add Rule**. The rule appears in the rule table to the right. Click **Apply Rule Changes** to apply the rule change. The security group now allows port 3333 access from anywhere.

**AWS Management Console** > **Amazon Virtual Private Cloud (VPC)**

Region: US East (Virginia) | VPC: All VPCs

**Security Groups**

Viewing: VPC Security Groups

Group ID	Name	VPC	Description
sg-bc100dd0	default	vpc-8388b9ea (10.0.0.0/16)	default VPC security group

**1 Security Group selected**

Security Group: default

Details | **Inbound** | Outbound

Create a new rule: Custom TCP rule

Port range:

Source:

Port (Service)	Source	Action
ALL	sg-bc100dd0	Delete
TCP Port (Service)	Source	Action
3333	0.0.0.0/0	Delete

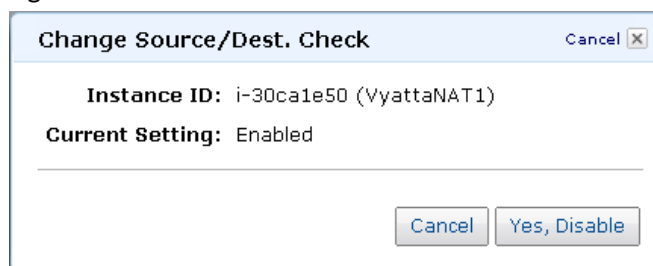
## Allow the Instance to Be Used for NAT

In order for the instance to be used as a NAT device, source and destination address checking must be disabled.

To disable source and destination address checking:

- 1 Select **Amazon Elastic Compute Cloud (EC2)** on the dropdown at the top of the AWS Management Console.
- 2 In the left navigation pane, select **Instances**. The **My Instances** page opens.
- 3 Right-click the row containing the Vyatta NAT1 instance. Select **Change Source / Dest Check** from the right-click menu. The **Change Source / Dest. Check** dialog opens.

Figure 2-1

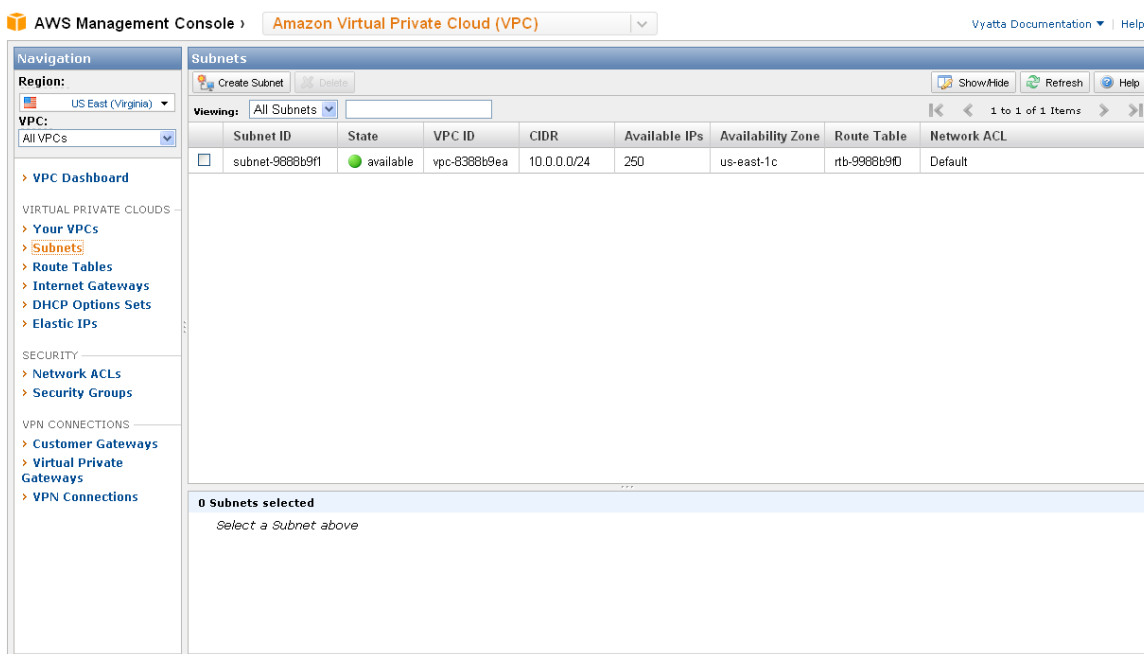


- 4 Make sure that **Current Setting:** is set to **Enabled**. Click **Yes, Disable**. The instance no longer checks source and destination address.

## Create a Private Subnet

Create a new subnet within the VPC. This subnet is made to be private in a later step.

- 1 Select **Amazon Virtual Private Cloud (VPC)** on the dropdown at the top of the AWS Management Console.
- 2 On the left navigation pane, select **Subnets**. The **Subnets** page opens.



3 Click **Create Subnet**. The **Create Subnet** dialog opens.

The **Create Subnet** dialog box contains the following information:

Please use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Please note that block sizes must be between a /16 netmask and /28 netmask. You can create no more than 20 subnets per VPC. Also, please note that a subnet can be the same size as your VPC.

**VPC:** vpc-8388b9ea (10.0.0.0/16)

**Availability Zone:** No Preference

**CIDR Block:** 10.0.1.0/24 (e.g., 10.0.0.0/24)

Buttons: **Cancel**, **Yes, Create**

4 In the **CIDR Block:** field, enter **10.0.1.0/24** and click **Yes, Create**.

This subnet must be within the 10.0.0.0/16 range that was defined for the VPC, but outside the 10.0.0.0/24 range configured for the public subnet.

The new subnet appears in the list of subnets.



**AWS Management Console** › Amazon Virtual Private Cloud (VPC)

Region: US East (Virginia) VPC: All VPCs

**Subnets**

Viewing: All Subnets

Subnet ID	State	VPC ID	CIDR	Available IPs	Availability Zone	Route Table	Network ACL
subnet-9688b9f1	available	vpc-8388b9ea	10.0.0.0/24	250	us-east-1c	rtb-9688b9f0	Default
subnet-4375452a	available	vpc-8388b9ea	10.0.1.0/24	251	us-east-1c	rtb-8588b9ec	Default

1 Subnet selected

Subnet: subnet-4375452a

CIDR: 10.0.1.0/24 VPC: vpc-8388b9ea Availability Zone: us-east-1c

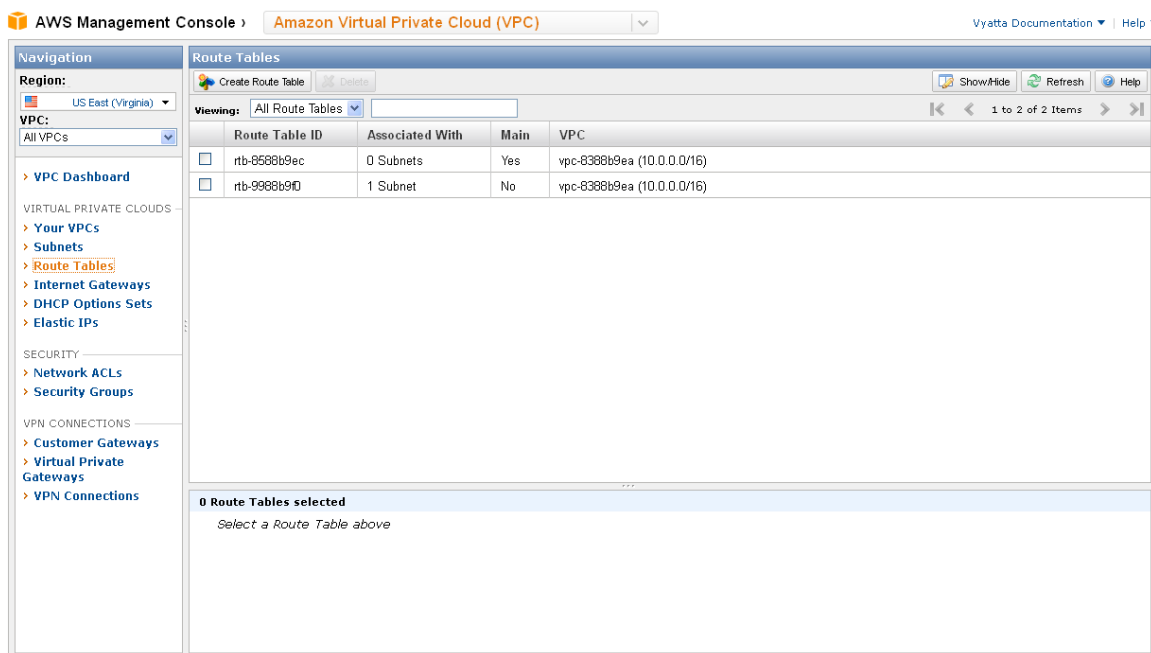
Route Table: rtb-8588b9ec (replace)

Destination	Target
10.0.0.0/16	local

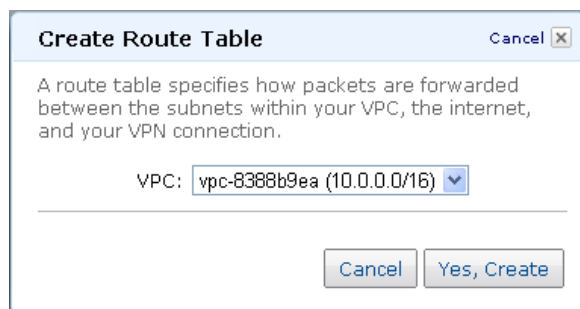
## Create a Route Table for the Private Subnet

This step enables access to instances within the private subnet in the VPC, and access from the private subnet to the Internet through the newly-created Vyatta NAT device.

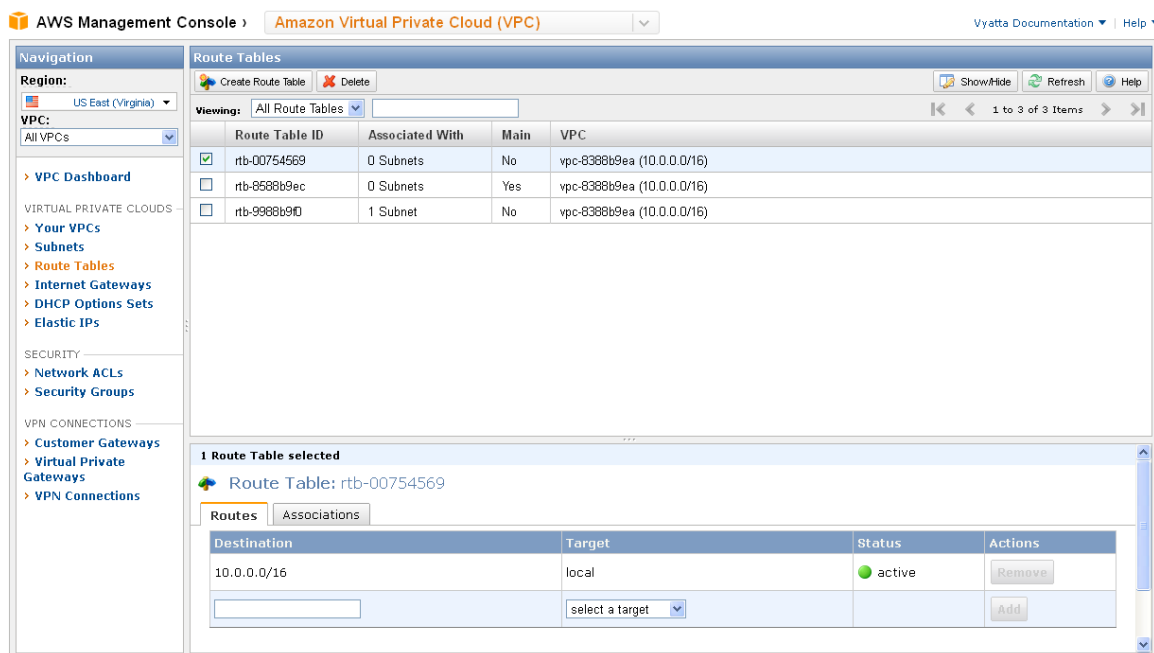
- 1 Select **Amazon Virtual Private Cloud (VPC)** on the dropdown at the top of the AWS Management Console.
- 2 In the left navigation pane, select **Route Tables**. The **Route Tables** page opens.



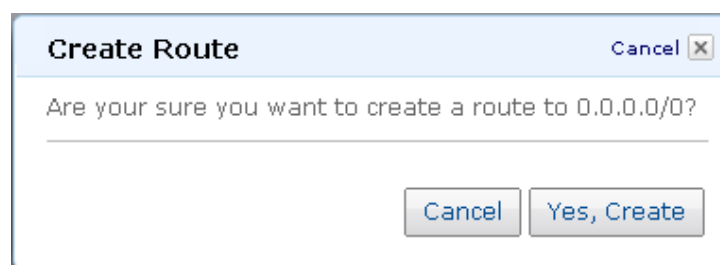
- 3 Click **Create Route Table**. The **Create Route Table** dialog opens.



- 4 In the **VPC:** field, select the VPC with subnet 10.0.0.0/16 and Click **Yes, Create**. The new route table appears in the list of route tables.



- 5 Select the new route table by clicking the checkbox to the left of it. Route table details appear at the bottom of the page. The default route allows access to other instances within the VPC.
- 6 Select the **Routes** tab.
- 7 In the field at the bottom of the **Destination** column, enter 0.0.0.0/0.
- 8 From the drop-down list at the bottom of the **Target** column, select the instance ID associated with the VyattaNAT1 instance. Press the **Add** button. The **Create Route** dialog appears.



- 9 Click **Yes, Create**. The new route is added to the table.

**AWS Management Console** > **Amazon Virtual Private Cloud (VPC)**

Region: US East (Virginia) | VPC: All VPCs

**Route Tables**

Viewing: All Route Tables

Route Table ID	Associated With	Main	VPC
<input checked="" type="checkbox"/> rtb-00754569	0 Subnets	No	vpc-8388b9ea (10.0.0.0/16)
<input type="checkbox"/> rtb-8588b9ec	0 Subnets	Yes	vpc-8388b9ea (10.0.0.0/16)
<input type="checkbox"/> rtb-9988b9d0	1 Subnet	No	vpc-8388b9ea (10.0.0.0/16)

**1 Route Table selected**

Route Table: rtb-00754569

**Routes** | Associations

Destination	Target	Status	Actions
10.0.0.0/16	local	active	<a href="#">Remove</a>
0.0.0.0/0	i-30ca1e50	active	<a href="#">Remove</a>
<input type="text"/>	<a href="#">select a target</a>		<a href="#">Add</a>

**10** Select the Associations tab. From the drop-down list, select the 10.0.1.0/24 subnet.

**AWS Management Console** > **Amazon Virtual Private Cloud (VPC)**

Region: US East (Virginia) | VPC: All VPCs

**Route Tables**

Viewing: All Route Tables

Route Table ID	Associated With	Main	VPC
<input checked="" type="checkbox"/> rtb-00754569	0 Subnets	No	vpc-8388b9ea (10.0.0.0/16)
<input type="checkbox"/> rtb-8588b9ec	0 Subnets	Yes	vpc-8388b9ea (10.0.0.0/16)
<input type="checkbox"/> rtb-9988b9d0	1 Subnet	No	vpc-8388b9ea (10.0.0.0/16)

**1 Route Table selected**

Route Table: rtb-00754569

**Routes** | **Associations**

Subnet	Actions
<a href="#">subnet-4375452a (10.0.1.0/24)</a>	<a href="#">Associate</a>

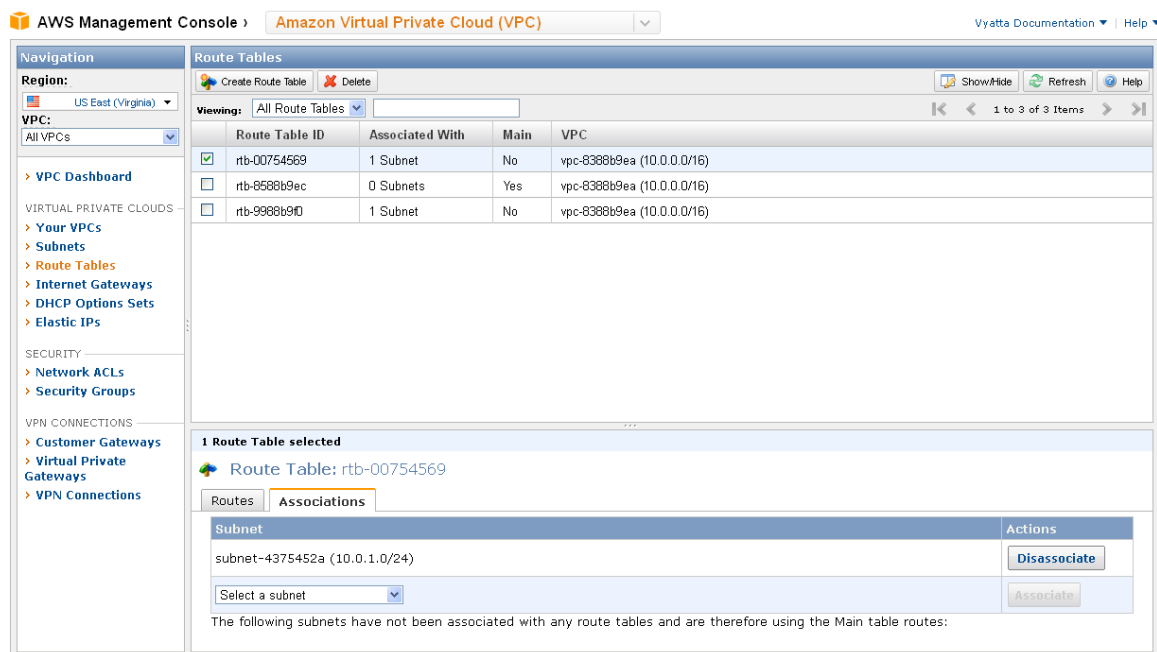
The following subnets have not been associated with any route tables and are therefore using the Main table routes:

- subnet-4375452a (10.0.1.0/24)

**11** Click Associate. The Associate Route Table dialog opens.



**12** Click **Yes, Associate**. The route table is associated with the 10.0.1.0/24 subnet.

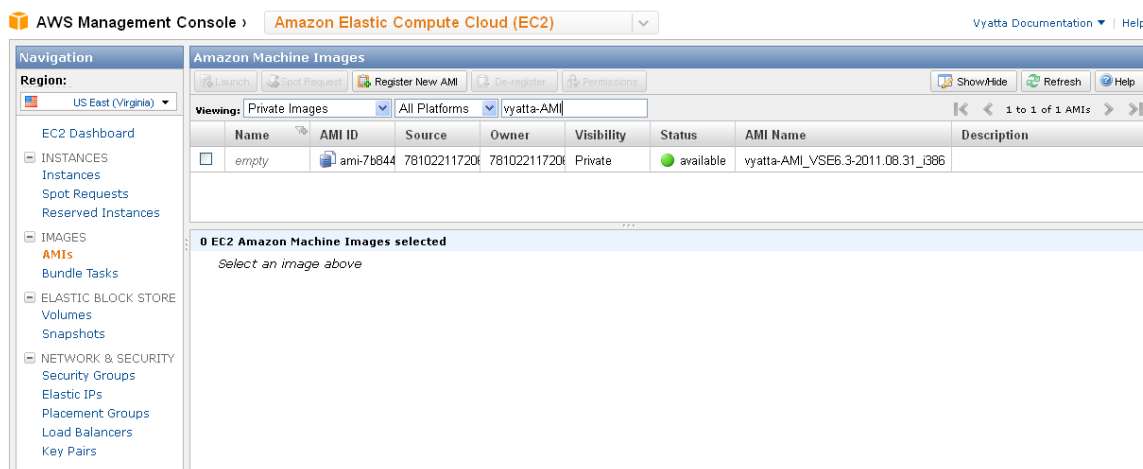


## Launch an Instance into the Private Subnet

Now that the private subnet 10.0.1.0/24 has been defined, we can launch an instance into it. Although the example launches another Vyatta AMI instance, any instance type could be launched.

**To launch a Vyatta AMI instance into the private subnet**

- 1** Select **Amazon Elastic Compute Cloud (EC2)** on the dropdown at the top of the AWS Management Console.
- 2** In the left navigation pane, select **AMIs**. The **Amazon Machine Images** page opens on the right.



- 3 In the **Viewing:** field, select **Private Images**, **All Platforms** and specify **vyatta-AMI** as the search string. Vyatta AMIs are listed.
- 4 Select a Vyatta AMI and click **Launch** at the top of the **Amazon Machine Images** page. The **Request Instances Wizard** starts at the **Instance Details** step.

Figure 2-2 Instance details

- 5 Select **Small (m1.small, 1.7GB)** as the **Instance Type**.

**NOTE** If you select **Micro (t1.micro, 613 MB)** you will not be able to launch the instance into your VPC.

- 6 Select **Launch Instances Into Your Virtual Private Cloud**.

- 7 In the **Subnet ID:** field, select the 10.0.1.0/24 subnet for attaching the instance to and click **Continue**. The **Advanced Instance Options** page opens.

**Request Instances Wizard** Cancel X

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

Number of Instances: 1  
Availability Zone: No Preference

**Advanced Instance Options**

Here you can choose a specific kernel or RAM disk to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default RAM Disk ID: Use Default

Monitoring: ☐ Enable CloudWatch detailed monitoring for this instance (additional charges will apply)

User Data:   
☒ as text   
☐ as file ☐ base64 encoded

IP Address: 10.0.1.20  
Optionally specify the IP address of your instance within the 10.0.1.0/24 subnet.

Termination Protection: ☐ Prevention against accidental termination.

Shutdown Behavior: Stop Choose the behavior when the instance is shutdown from within the instance.

Tenancy: Default

Back Continue

- 8 In the **IP Address:** field, enter 10.0.1.20 and press **Continue**. The **Add Tags** page appears.

**Request Instances Wizard** Cancel X

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. For example, you could define a tag with key = Name and value = Webserver. You can add up to 10 unique keys to each instance along with an optional value for each key. For more information, go to [Using Tags](#) in the *EC2 User Guide*.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
Name	VyattaPrivate	X
		X

Add another Tag. (Maximum of 10)

Back Continue

- 9 In the **Value** column to the right of the **Name** key, enter **VyattaPrivate** and click **Continue**. The **Create Key Pair** page opens.

The screenshot shows the 'Request Instances Wizard' window. The progress bar at the top indicates the current step is 'CREATE KEY PAIR'. Below the progress bar, there is a text box explaining that public/private key pairs allow secure connection to the instance. Below this, there are three radio button options: 'Choose from your existing Key Pairs' (which is selected), 'Create a new Key Pair', and 'Proceed without a Key Pair'. Under the selected option, there is a dropdown menu labeled 'Your existing Key Pairs:' with 'testkey' selected. At the bottom of the window, there are 'Back' and 'Continue' buttons.

- 10 Select **Choose from your existing Key Pairs** and select an existing key pair from the **Your existing Key Pairs** drop-down list. Click **Continue**. The **Configure Firewall** page opens.

Figure 2-3 Configure firewall

The screenshot shows the 'Request Instances Wizard' window, now at the 'CONFIGURE FIREWALL' step. The progress bar at the top reflects this change. Below the progress bar, there is a text box explaining that security groups determine whether a network port is open or blocked. Below this, there are two radio button options: 'Choose one or more of your existing Security Groups' (which is selected) and 'Create a new Security Group'. Under the selected option, there is a list box showing 'sg-bc100dd0 - default' as the selected group. Below the list box, it says '(Selected groups: sg-bc100dd0)'. At the bottom of the window, there are 'Back' and 'Continue' buttons.

- 11 Select the default security group and click **Continue**. The **Review** page opens.



**Request Instances Wizard**

CHOOSE AN AMI | INSTANCE DETAILS | CREATE KEY PAIR | CONFIGURE FIREWALL | REVIEW

Please review the information below, then click **Launch**.

**AMI:** Other Linux AMI ID ami-7b844712 (i386) [Edit AMI](#)

---

**Number of Instances:** 1  
**VPC ID:** vpc-8388b9ea  
**VPC Subnet:** subnet-4375452a (10.0.1.0/24)  
**Availability Zone:** No Preference  
**Instance Type:** Small (m1.small)  
**Instance Class:** On Demand [Edit Instance Details](#)

---

**Monitoring:** Disabled **Termination Protection:** Disabled  
**Tenancy:** Default  
**Kernel ID:** Use Default **Shutdown Behavior:** Stop  
**RAM Disk ID:** Use Default  
**IP Address:** 10.0.1.20  
**User Data:** [Edit Advanced Details](#)

---

**Key Pair Name:** testkey [Edit Key Pair](#)

---

**Security Group(s):** sg-bc100dd0 [Edit Firewall](#)

[< Back](#) [Launch](#)

**12** Review the details for the instance you are creating. When you are satisfied, click **Launch**. The instance starts. Click **Close**.

**13** To view the status of the newly launched instance, select **Instances** on the left navigation pane within the **Amazon Elastic Compute Cloud (EC2)**.

**AWS Management Console** | **Amazon Elastic Compute Cloud (EC2)** | [Vyatta Documentation](#) | [Help](#)

**Navigation**

Region: **US East (Virginia)**

- EC2 Dashboard
- INSTANCES
  - Instances**
  - Spot Requests
  - Reserved Instances
- IMAGES
  - AMIs
  - Bundle Tasks
- ELASTIC BLOCK STORE
  - Volumes
  - Snapshots
- NETWORK & SECURITY
  - Security Groups
  - Elastic IPs
  - Placement Groups
  - Load Balancers
  - Key Pairs

**My Instances**

[Launch Instance](#) [Instance Actions](#) [Show/Hide](#) [Refresh](#) [Help](#)

Viewings: **All Instances** | **All Instance Types** |

	Name	Instance	Root Device	Type	Status	Security Groups	Elastic IP	Private IP
<input type="checkbox"/>	VyattaNAT1	i-30ca1e50	ebs	m1.small	running	default	184.72.119.76	10.0.0.10
<input type="checkbox"/>	VyattaPrivate	i-ca6db9aa	ebs	m1.small	running	default	--	10.0.1.20

1 to 2 of 2 Instances

**0 EC2 Instances selected**

Select an instance above

## Access the Private Instance Remotely

Since the default security group is associated with the instance, remote SSH connections will be allowed through to it.

### To access the instance remotely using SSH

- 1 On a remote machine, open an SSH session. As the destination, use the Elastic IP address you associated with the Vyatta NAT instance. Specify 3333 as the port.

The Vyatta NAT device has been configured to translate any connections to port 3333 to address 10.0.1.20 port 22. This connection is routed to the instance created within the private subnet.

**NOTE** On Linux/UNIX systems use the **ssh** command. On Windows machines use a program such as **putty** for SSH access.

- 2 Once connected you will see the **login as:** prompt. Log on to the instance using the default credentials: user **vyatta** with password **vyatta**.

## Verify the Instance is Working as Expected

Once you are logged into the system, issue the following commands to confirm that it is working as expected.

- 1 Confirm the IP address that is associated with the Ethernet interface.

```
vyatta@vyatta:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface      IP Address      S/L  Description
-----
eth0           10.0.1.20/24    u/u
lo             127.0.0.1/8     u/u
              ::1/128
```

- 2 Confirm the information that has been provided by the Amazon DHCP server.

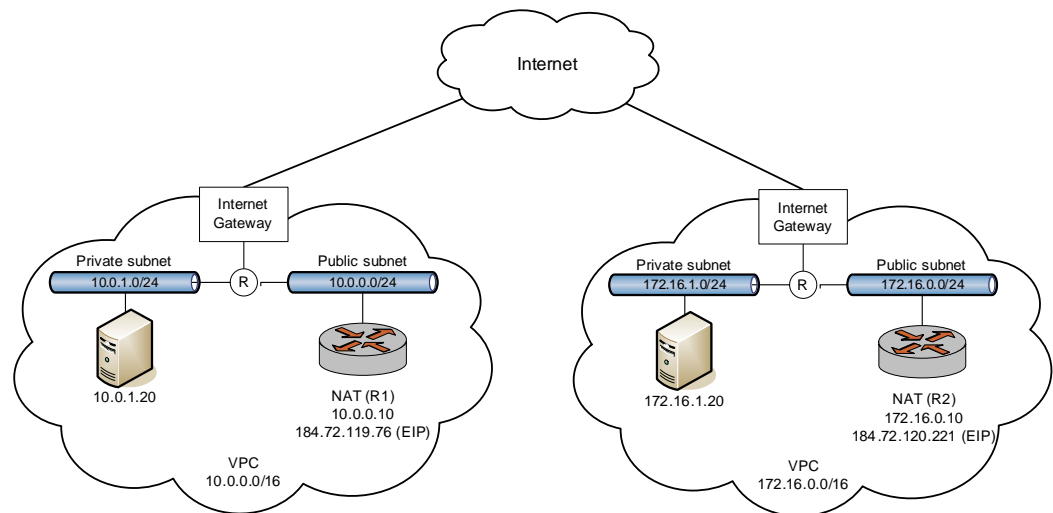
```
vyatta@vyatta:~$ show dhcp client leases
interface : eth0
ip address : 10.0.1.20 [Active]
subnet mask: 255.255.255.0
router     : 10.0.1.1
name server: 10.0.0.2
dhcp server: 10.0.1.1
lease time : 3600
last update: Wed Aug 31 19:25:23 GMT 2011
expiry     : Wed Aug 31 20:25:23 GMT 2011
reason     : RENEW
vyatta@vyatta:~$
```

- 3 Confirm that the instance has access to the Internet using **ping** (press <Ctrl>+c to stop the output).

```
vyatta@vyatta:~$ ping www.vyatta.com
PING www.vyatta.com (76.74.103.45) 56(84) bytes of data.
64 bytes from www.vyatta.com (76.74.103.45): icmp_req=1 ttl=46 time=74.4
ms
64 bytes from www.vyatta.com (76.74.103.45): icmp_req=2 ttl=46 time=74.5
ms
^C
--- www.vyatta.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 74.492/74.502/74.513/0.273 ms
vyatta@vyatta:~$ ^C
vyatta@vyatta:~$
```

## Creating a Site-to-site IPsec VPN Connection

In this example, a site-to-site IPsec VPN connection is created between the NAT devices in separate VPCs. It assumes that Vyatta NAT instances and instances within private subnets have been created within the VPCs according to the steps in [“Creating a NAT Device” on page 18](#). The following diagram shows the configuration.



To allow inbound Internet Key Exchange (IKE), Encapsulating Security Payload (ESP), IPsec NAT-T, and ping traffic, add four rules to the default VPC security group in each VPC. The first inbound rule (for IKE) allows UDP traffic on port 500 from any source (0.0.0.0/0). The second inbound rule (for ESP) is a **Custom protocol rule** and allows IP protocol 50 traffic from any source (0.0.0.0/0). The third inbound rule

(for IPsec NAT-T) allows UDP traffic on port 4500 from any source (0.0.0.0/0). The fourth inbound rule (for ping traffic) allows ICMP traffic from any source (0.0.0.0/0). See [“Modify the Default Security Group” on page 20](#) as a reference.

To provide an IPsec VPN endpoint on the NAT device R1, configure it as follows:

```
vyatta@R1# show vpn
ipsec {
    esp-group ESP-1W {
        compression disable
        lifetime 1800
        mode tunnel
        pfs enable
        proposal 1 {
            encryption aes256
            hash sha1
        }
        proposal 2 {
            encryption 3des
            hash md5
        }
    }
    ike-group IKE-1W {
        lifetime 3600
        proposal 1 {
            encryption aes256
            hash sha1
        }
        proposal 2 {
            encryption aes128
            hash sha1
        }
    }
    ipsec-interfaces {
        interface eth0
    }
    nat-networks {
        allowed-network 0.0.0.0/0 {
            exclude 10.0.0.0/16
        }
    }
    nat-traversal enable
    site-to-site {
        peer 184.72.120.221 {
            authentication {
                id @R1
                mode pre-shared-secret
            }
        }
    }
}
```

```
        pre-shared-secret test_key_1
        remote-id @R2
    }
    connection-type initiate
    default-esp-group ESP-1W
    ike-group IKE-1W
    local-ip 10.0.0.10
    tunnel 1 {
        allow-nat-networks disable
        allow-public-networks disable
        local {
            subnet 10.0.0.0/16
        }
        remote {
            subnet 172.16.0.0/16
        }
    }
}
}
[edit]
vyatta@R1#
```

---

To provide an IPsec VPN endpoint on the NAT device R2, configure it as follows:

---

```
vyatta@R2# show vpn
ipsec {
    esp-group ESP-1E {
        compression disable
        lifetime 1800
        mode tunnel
        pfs enable
        proposal 1 {
            encryption aes256
            hash sha1
        }
        proposal 2 {
            encryption 3des
            hash md5
        }
    }
    ike-group IKE-1E {
        lifetime 3600
        proposal 1 {
            encryption aes256
```

```
        hash sha1
    }
    proposal 2 {
        encryption aes128
        hash sha1
    }
}
ipsec-interfaces {
    interface eth0
}
nat-networks {
    allowed-network 0.0.0.0/0 {
        exclude 172.16.0.0/16
    }
}
nat-traversal enable
site-to-site {
    peer 184.72.119.76 {
        authentication {
            id @R2
            mode pre-shared-secret
            pre-shared-secret test_key_1
            remote-id @R1
        }
        connection-type initiate
        default-esp-group ESP-1E
        ike-group IKE-1E
        local-ip 172.16.0.10
        tunnel 1 {
            allow-nat-networks disable
            allow-public-networks disable
            local {
                subnet 172.16.0.0/16
            }
            remote {
                subnet 10.0.0.0/16
            }
        }
    }
}
}
[edit]
vyatta@R2#
```

---

Test the configuration by pinging a device in one private subnet (10.0.1.20) from a device in the other private subnet (172.16.1.20).

```

vyatta@vyatta:~$ ping 10.0.1.20
PING 10.0.1.20 (10.0.1.20) 56(84) bytes of data.
64 bytes from 10.0.1.20: icmp_req=1 ttl=64 time=0.439 ms
64 bytes from 10.0.1.20: icmp_req=2 ttl=64 time=0.572 ms
64 bytes from 10.0.1.20: icmp_req=3 ttl=64 time=0.430 ms
64 bytes from 10.0.1.20: icmp_req=4 ttl=64 time=0.448 ms
^C
--- 10.0.1.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.430/0.472/0.572/0.059 ms
vyatta@vyatta:~$

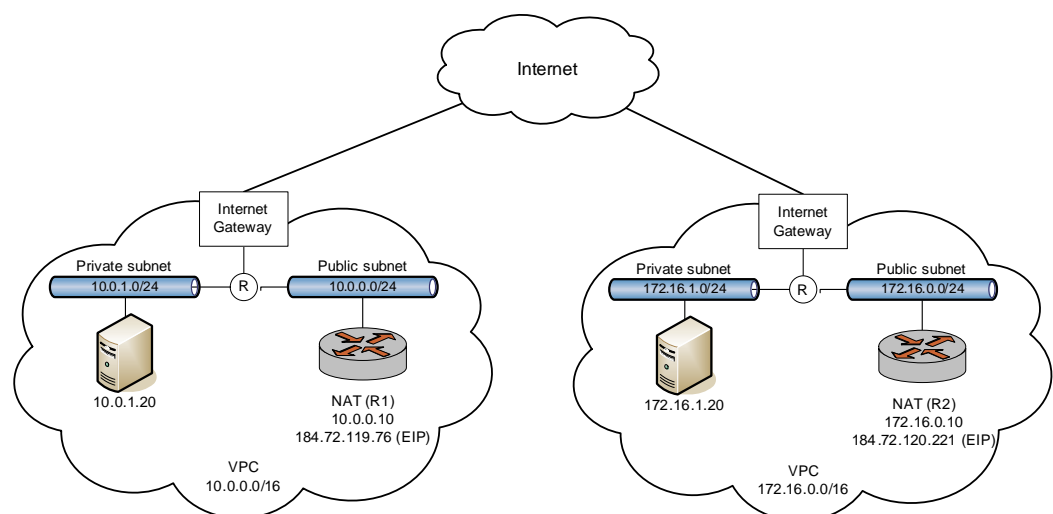
```

While this example shows a site-to-site IPsec VPN connection between sites in two different VPCs, the sites can also be located in non-VPC locations (for example, a branch office or a data center).

For further information on IPsec VPN configuration, please see the *Vyatta VPN Reference Guide*.

## Creating a Site-to-site OpenVPN Connection

In this example, a site-to-site OpenVPN connection is created between the NAT devices in separate VPCs. It assumes that Vyatta NAT instances and instances within private subnets have been created within the VPCs according to the steps in [“Creating a NAT Device” on page 18](#). The following diagram shows the configuration.



To allow inbound OpenVPN and ping traffic, add two rules to the default VPC security group in each VPC. The first inbound rule allows TCP traffic on port 1194 from any source (0.0.0.0/0). The second inbound rule allows ICMP traffic from any source (0.0.0.0/0). See [“Modify the Default Security Group” on page 20](#) as a reference.

To provide an OpenVPN endpoint on the NAT device R1, configure it as follows:

---

```
vyatta@R1# show interfaces openvpn
openvpn vtun0 {
    local-address 192.168.200.1 {
    }
    mode site-to-site
    remote-address 192.168.200.2
    remote-host 184.72.120.221
    shared-secret-key-file /config/auth/secret
}
[edit]
vyatta@R1#
```

---

**NOTE** The shared secret key file is created using **generate vpn openvpn <filename>** and then copied to both systems.

To provide an OpenVPN endpoint on the NAT device R2, configure it as follows:

---

```
vyatta@R2# show interfaces openvpn
openvpn vtun0 {
    local-address 192.168.200.2 {
    }
    mode site-to-site
    remote-address 192.168.200.1
    remote-host 184.72.119.76
    shared-secret-key-file /config/auth/secret
}
[edit]
vyatta@R2#
```

---

Test the configuration by pinging a device in one private subnet (10.0.1.20) from a device in the other private subnet (172.16.1.20).

---

```
vyatta@vyatta:~$ ping 10.0.1.20
PING 10.0.1.20 (10.0.1.20) 56(84) bytes of data.
```



```

64 bytes from 10.0.1.20: icmp_req=1 ttl=64 time=0.439 ms
64 bytes from 10.0.1.20: icmp_req=2 ttl=64 time=0.572 ms
64 bytes from 10.0.1.20: icmp_req=3 ttl=64 time=0.430 ms
64 bytes from 10.0.1.20: icmp_req=4 ttl=64 time=0.448 ms
^C
--- 10.0.1.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.430/0.472/0.572/0.059 ms
vyatta@vyatta:~$

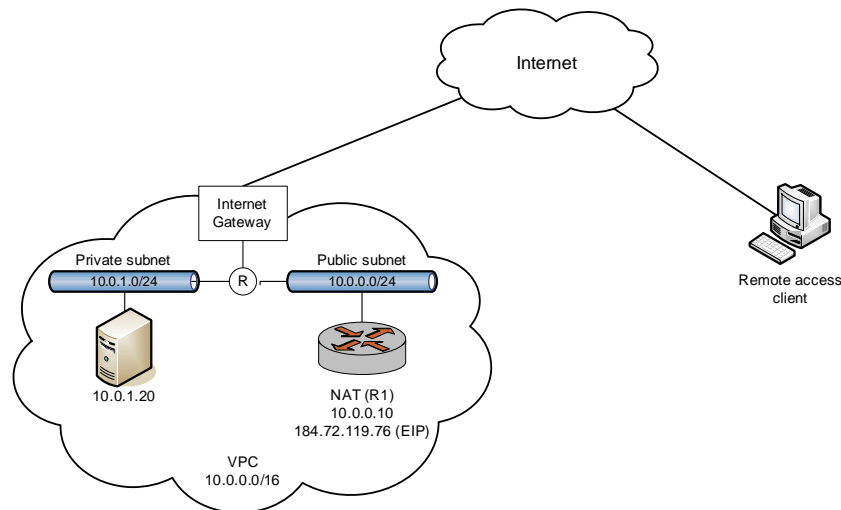
```

While this example shows a site-to-site OpenVPN connection between sites in two different VPCs, the sites can also be located in non-VPC locations (for example, a branch office or a data center).

For further information on OpenVPN configuration, please see the *Vyatta VPN Reference Guide*.

## Creating a Remote Access VPN Connection

In this example, a Remote Access VPN connection is created between a remote client and the NAT device in a VPC. It assumes that a Vyatta NAT instance and an instance within a private subnet have been created within the VPC according to the steps in [“Creating a NAT Device” on page 18](#). The following diagram shows the configuration.



To allow inbound Remote Access VPN and ping traffic, add two rules to the default VPC security group. The first inbound rule allows TCP traffic on port 1723 from any source (0.0.0.0/0). The second inbound rule allows ICMP traffic from any source (0.0.0.0/0). See [“Modify the Default Security Group” on page 20](#) as a reference.

To provide a remote access server on the NAT device R1, configure it as follows:

---

```
vyatta@R1# show vpn pptp
remote-access {
    authentication {
        local-users {
            username test {
                password test
            }
        }
        mode local
    }
    client-ip-pool {
        start 10.0.1.100
        stop 10.0.1.150
    }
    outside-address 10.0.0.10
}
[edit]
vyatta@R1#
```

---

To configure a PPTP VPN client on a Windows XP SP2 system (the remote access client in this example), use the Windows “New Connection Wizard,” as follows:

- 1** In Windows, select **Start > Control Panel > Network Connections**.
- 2** Click **Create a new connection**. The New Connection Wizard launches. Click **Next**.
- 3** Select **Connect to the network at my workplace**. Click **Next**.
- 4** Select **Virtual Private Network connection**. Click **Next**.
- 5** Enter a name for the connection; for example, “Vyatta-PPTP.” Click **Next**.
- 6** Select **Do not dial the initial connection**. Click **Next**.
- 7** Enter the Elastic IP address. Click **Next**.
- 8** Select **Do not use my smart card**. Click **Next**.
- 9** Click **Finish**.

To connect to the VPN server, double-click the VPN connection icon, enter your user name (“test” in the example) and password (“test” in the example), and then click **Connect**. You can use the **show interfaces** and **show vpn remote-access** operational commands on the Vyatta VPN server to display the connected user on an interface named “pntpX,” where X is an integer.

**NOTE** You must make sure that nothing is blocking packets with protocol GRE or TCP port 1723 between the remote client and the VPN server. (Check firewall settings, home gateway, DSL modem, ISP, and so on.)

Test the configuration by pinging a device in the private network from the remote client (in this case, from the command line of the Windows client).

---

```
C:\> ping 10.0.1.20
```

```
Pinging 10.0.1.20 with 32 bytes of data:
```

```
Reply from 10.0.1.20: bytes=32 time=1ms TTL=64
Reply from 10.0.1.20: bytes=32 time<1ms TTL=64
Reply from 10.0.1.20: bytes=32 time<1ms TTL=64
Reply from 10.0.1.20: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 10.0.1.20:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

```
C:\ >
```

---

While this example shows a remote access VPN connection, OpenVPN can also be configured for remote access connections.

For further information on Remote Access VPN configuration, please see the *Vyatta VPN Reference Guide*.

## Chapter 3: Upgrading the System

This chapter explains how to upgrade Vyatta system software on a Vyatta AMI.

In this chapter:

- [Release-Specific Upgrade Information](#)
- [Upgrading in a Virtual Environment](#)

## Release-Specific Upgrade Information

---

Your system may have special upgrade considerations, depending on your release.

For release-specific upgrade information, and to ensure that configuration information is correctly preserved, see the Release Notes for your release.

Before upgrading, save your existing configuration file for reference. Your configuration file is named **config.boot** and is located in the directory **/config**.

## Upgrading in a Virtual Environment

---

Upgrading in a virtual environment involves two steps:

- 1 Install a fresh virtualized environment.
- 2 Migrate your configuration.

### Install the new virtualized Vyatta system

Install a fresh virtualized Vyatta system. To do this, following the instructions in [Chapter 1: Installing the System](#).

### Migrate your configuration

- 1 In configuration mode on the old system, use the **save** command to save the current configuration.
- 2 For all Ethernet interfaces, remove the hardware ID values using the **delete interfaces ethernet ethx hw-id** command to remove the hardware ID values, then **commit** and **save** the configuration to a name other than **config.boot** (for example, **save oldconfig**).
- 3 Use the **load** command to return the original configuration to the old system.
- 4 Use the **set service ssh** command and then the **commit** command to configure the system to allow for SCP file transfer.
- 5 In configuration mode on the new system, assign an IP address to an interface residing on the same subnet as one on the old system (for example, **set interfaces ethernet eth0 address 192.168.1.99/24**), and then commit the change.
- 6 Copy the saved configuration (the one with the hardware IDs removed) from the old system to the new system. For example, if the old system is at 192.168.1.20, the saved configuration file is name **oldconfig**, and the username **vyatta** is available on the old system, issue the **scp** command as follows:

```
scp vyatta@192.168.1.20:/config/oldconfig /config/oldconfig
```

- 7 Load the copied configuration using the **load** command (for example, **load oldconfig**). At this point, the configuration on the new system should match that on the old system (except for the hardware IDs).
- 8 Shut down the old system using the **shutdown** command.

# Glossary

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point

---

DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP security
IPv4	IP Version 4
IPv6	IP Version 6
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol

---



---

LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit

---

---

QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
SSID	Service Set Identifier
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol

---

---

WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access

---