VYATTA, INC. | Vyatta System

AMI

INSTALLING THE SYSTEM/USER GUIDE

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

COPYRIGHT

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.

PROPRIETARY NOTICES

RELEASE DATE: October 2012

DOCUMENT REVISION. 6.5R1 v01

RELEASED WITH: 6.5R1

PART NO. A0-0243-10-0005

Contents

Preface

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

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

This document explains how to obtain the Vyatta AMI and launch it into a Virtual Private Cloud (VPC) within the AWS cloud, and then how 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:

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.
`bold Monospace`	Your input: something you type at a command line.
bold	Commands, keywords, and file names, when mentioned inline. Objects in the user interface, such as tabs, buttons, screens, and panes.
italics	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 and 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

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 image. You instantiate a copy of the image 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. You can obtain the Vyatta AMI from Amazon's AWS Marketplace.

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 obtain and 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:

Amazon Web Services Account

- 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.

Amazon Web Services Skills

- You have mastered general AWS skills, including the following:
  - — 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)

Learning About AWS

It is beyond the scope of this guide to describe how to use AWS. Before trying to use a Vyatta AMI with AWS, review the AWS documentation shown in Table 1-1.

Table 1-1 Amazon Web Services Reference Documentation

What	Where
AWS	
Introduction to AWS webinar in the Solutions playlist	http://aws.amazon.com/resources/webinars/
AWS documentation library	http://aws.amazon.com/documentation/
Amazon EC2	
Amazon EC2 documentation index	http://aws.amazon.com/documentation/ec2/
Amazon EC2 Getting Started Guide	http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/
Amazon EC2 User Guide	http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/
Amazon VPC documentation index	http://aws.amazon.com/documentation/vpc/
Amazon VPC	
Amazon VPC Getting Started Guide	http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide/
Amazon VPC User Guide	http://docs.amazonwebservices.com/AmazonVPC/latest/UserGuide/

Installation Options

This document describes how to install a Vyatta AMI into a Virtual Private Cloud (VPC) within the AWS environment as this how it is most likely to be deployed. For evaluation purposes, it is possible to simply launch the Vyatta AMI directly into the EC2 infrastructure - not within a VPC. This simplifies installation as a number of the steps described below can be eliminated. To do this, go directly to "Obtaining the Vyatta AMI from the AWS Marketplace" on page 20 and perform only steps 1 to 5.

Creating a VPC

Before you obtain a Vyatta AMI you need to create a VPC that it can be launched into. 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 are logged in to the AWS Management Console and 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.

Modifying the Default Security Group

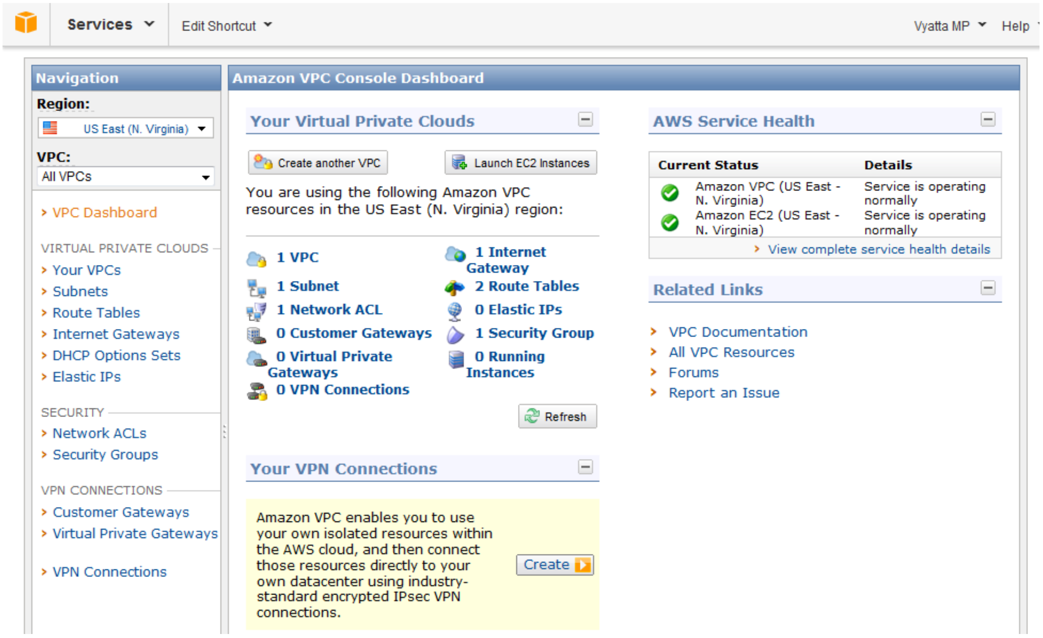
Security groups provide the policies that control traffic flow and access for EC2 instances and instances within aVPC. EC2 security groups and VPC security groups are independent of one another. EC2 security groups cannot be used for instances within a VPC, and VPC security groups cannot be used for 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 AMI instance(s) you'll be creating within the VPC. To provide remote SSH access into the VPC, either create a new security 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 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.



Services Edit Shortcut Vyatta MP Help

Navigation

Region: US East (N. Virginia)

VPC: All VPCs

- VPC Dashboard

VIRTUAL PRIVATE CLOUDS
- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- DHCP Options Sets
- Elastic IPs

SECURITY
- Network ACLs
- Security Groups

VPN CONNECTIONS
- Customer Gateways
- Virtual Private Gateways
- VPN Connections

Amazon VPC Console Dashboard

Your Virtual Private Clouds

Create another VPC Launch EC2 Instances

You are using the following Amazon VPC resources in the US East (N. Virginia) region:

1 VPC	1 Internet Gateway
1 Subnet	2 Route Tables
1 Network ACL	0 Elastic IPs
0 Customer Gateways	1 Security Group
0 Virtual Private Gateways	0 Running Instances
0 VPN Connections	

Refresh

Your VPN Connections

Amazon VPC enables you to use your own isolated resources within the AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.

Create

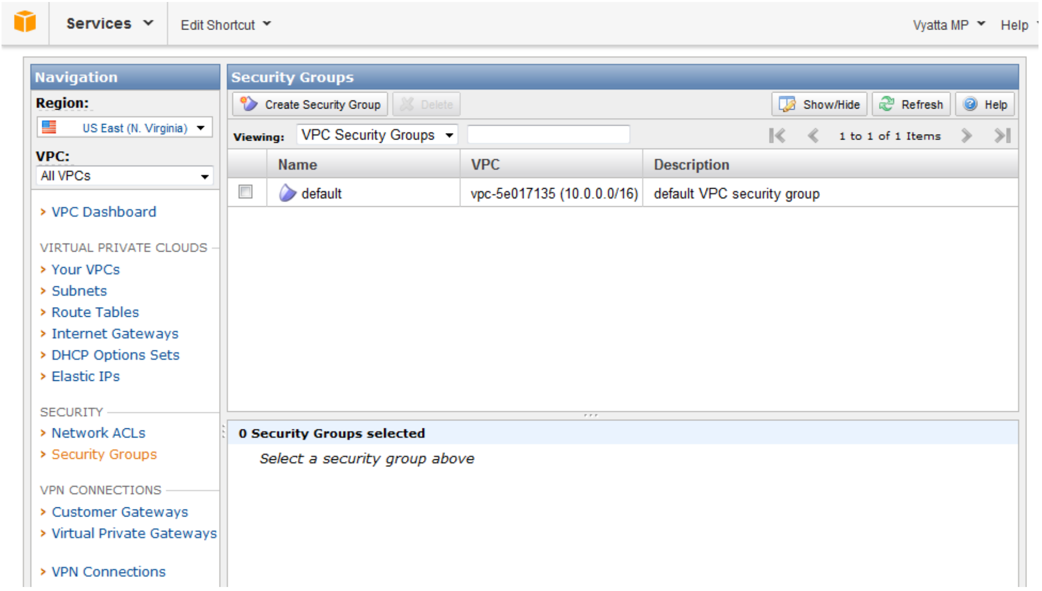
AWS Service Health

Current Status	Details
✓ Amazon VPC (US East - N. Virginia)	Service is operating normally
✓ Amazon EC2 (US East - N. Virginia)	Service is operating normally

View complete service health details

Related Links
- VPC Documentation
- All VPC Resources
- Forums
- Report an Issue

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



Services Edit Shortcut Vyatta MP Help

Navigation

Region: US East (N. Virginia)

VPC: All VPCs

- VPC Dashboard

VIRTUAL PRIVATE CLOUDS
- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- DHCP Options Sets
- Elastic IPs

SECURITY
- Network ACLs
- Security Groups

VPN CONNECTIONS
- Customer Gateways
- Virtual Private Gateways
- VPN Connections

Security Groups

Create Security Group Delete Show/Hide Refresh Help

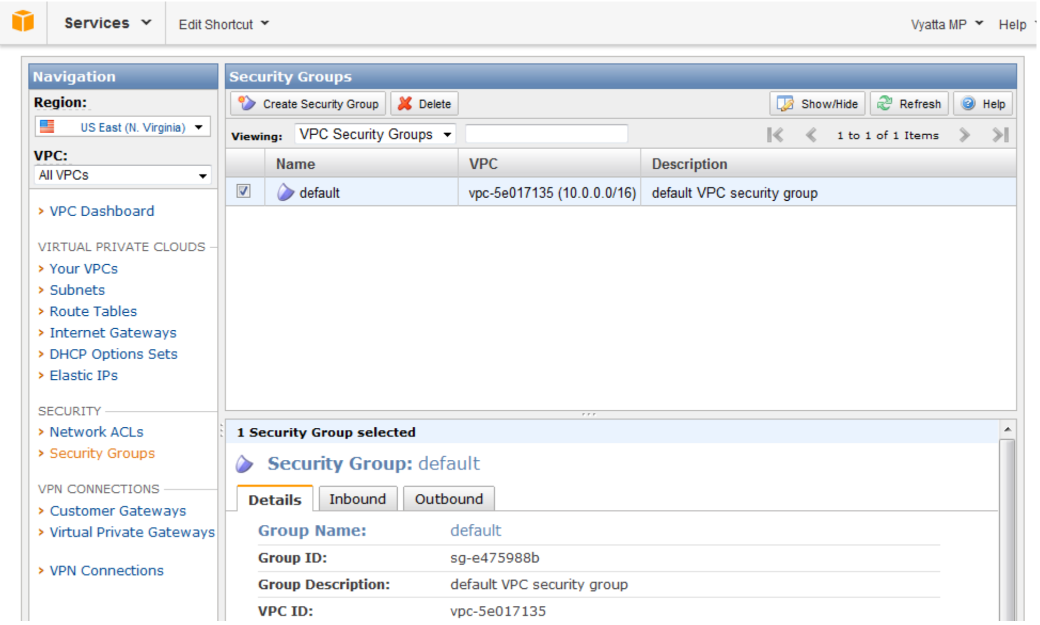
Viewing: VPC Security Groups 1 to 1 of 1 Items

	Name	VPC	Description
□	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

0 Security Groups selected

Select a security group above

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



	Name	VPC	Description
☑	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

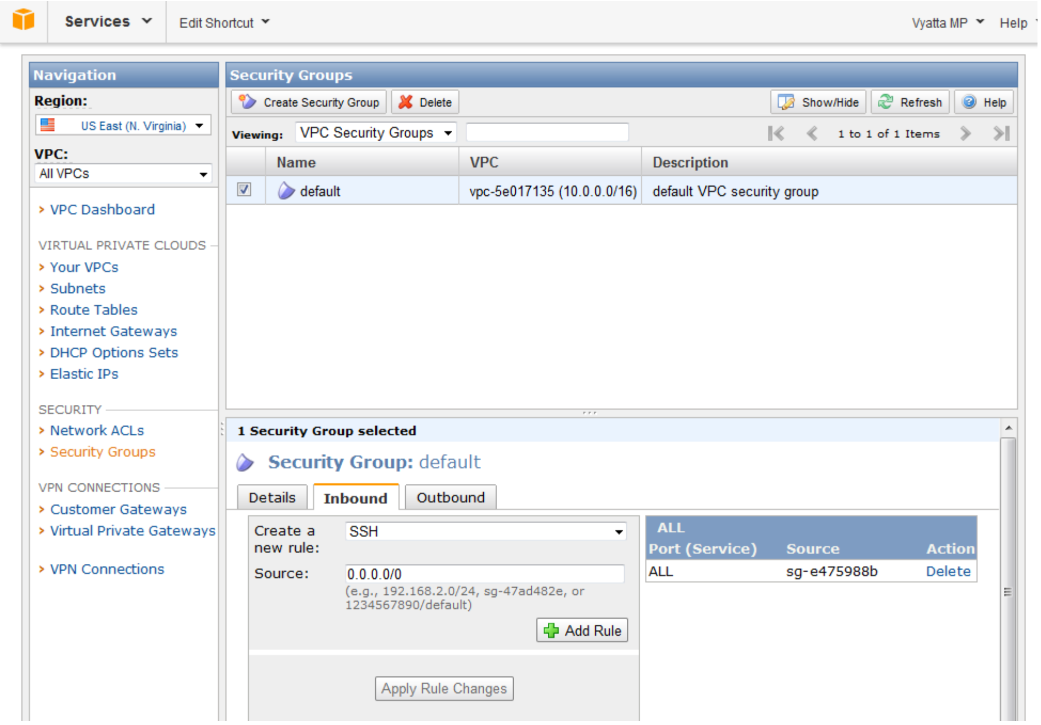
1 Security Group selected

Security Group: default

Group Name:	default
Group ID:	sg-e475988b
Group Description:	default VPC security group
VPC ID:	vpc-5e017135

4 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.



	Name	VPC	Description
☑	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

1 Security Group selected

Security Group: default

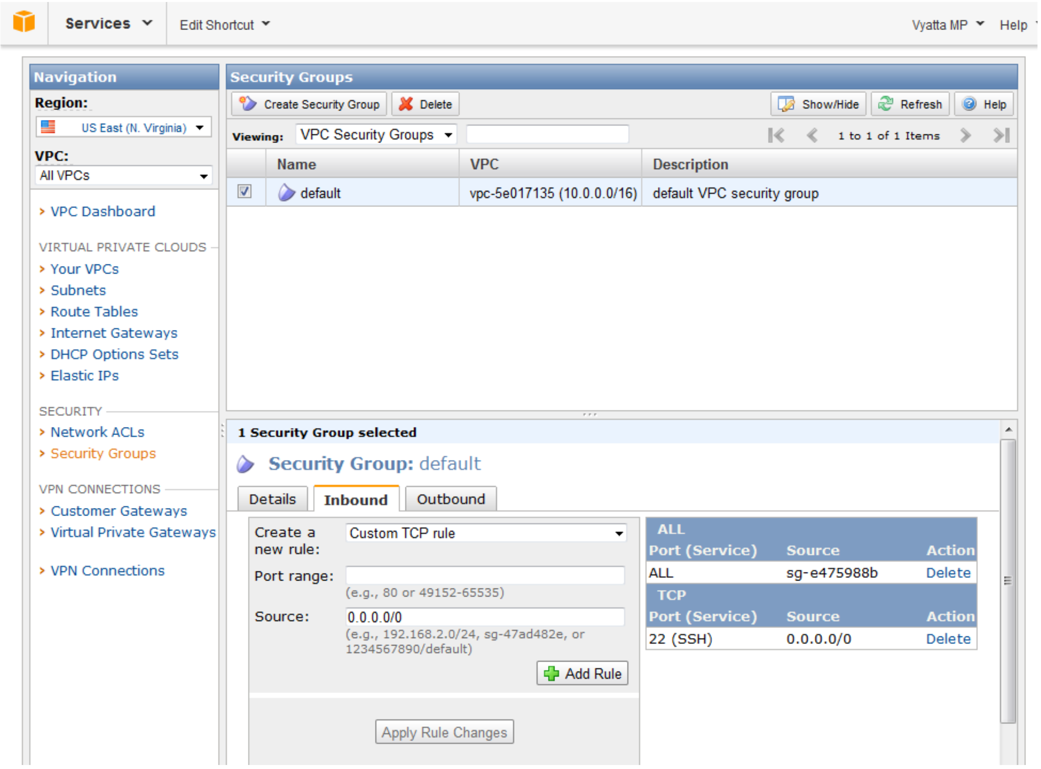
Create a new rule: SSH

Source: 0.0.0.0/0

(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

ALL		
Port (Service)	Source	Action
ALL	sg-e475988b	Delete

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.



	Name	VPC	Description
☑	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

Create a new rule: Custom TCP rule

Port range:

(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0

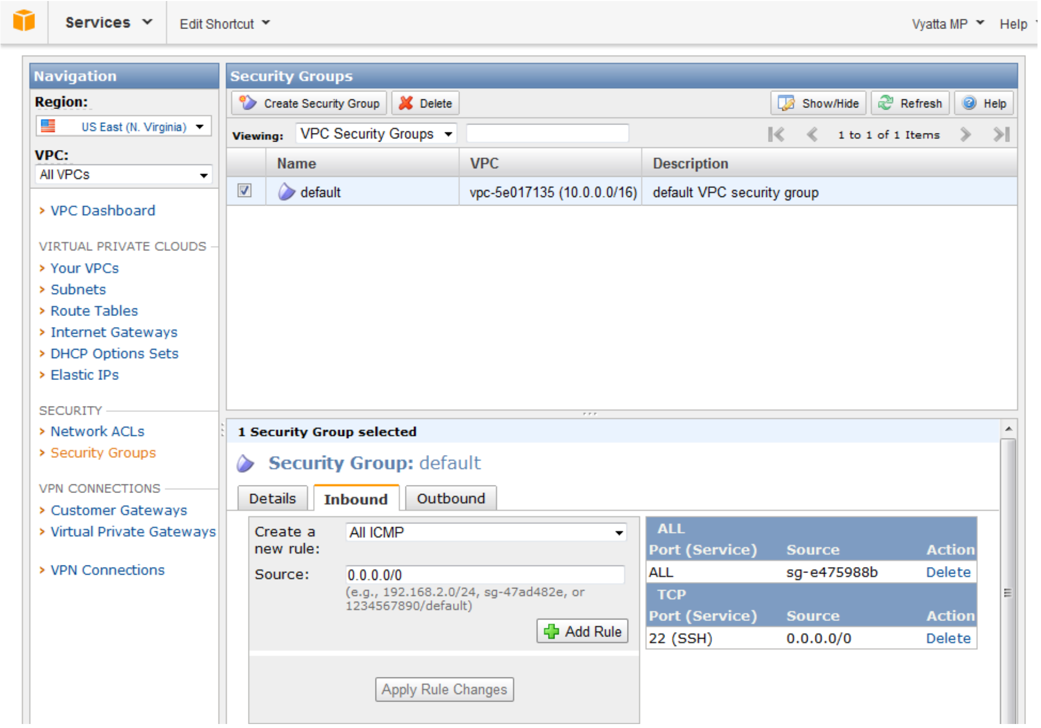
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

ALL		
Port (Service)	Source	Action
ALL	sg-e475988b	Delete
TCP		
Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete

The default VPC security group does not allow instances within the VPC to respond to pings (ICMP echo requests) from remote devices. In many cases this is desireable. For our testing purposes, it is desirable to determine that an instance is reachable so we want to allow ICMP traffic. This example modifies the default security group to allow incoming ICMP traffic from anywhere.

To modify the default security group to allow ICMP traffic

1 In the **Create a new rule:** field of the **Inbound** tab, select **All ICMP** from the drop-down menu.



Region: US East (N. Virginia)

VPC: All VPCs

Viewing: VPC Security Groups

	Name	VPC	Description
☑	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

1 Security Group selected

Security Group: default

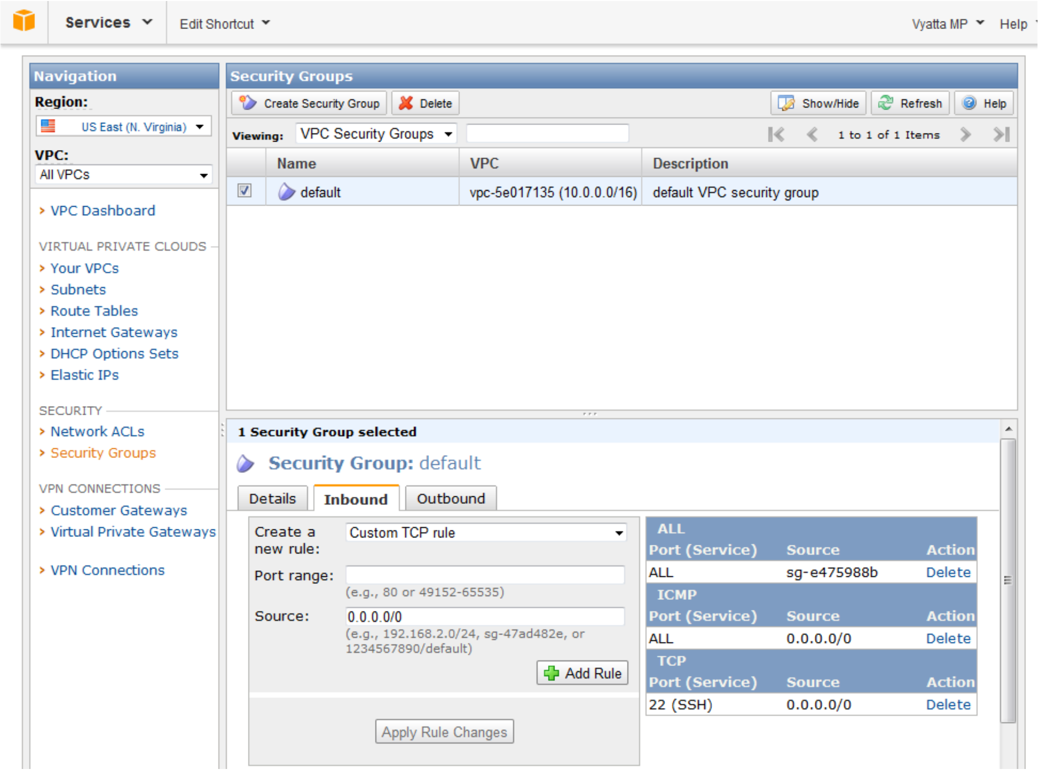
Create a new rule: All ICMP

Source: 0.0.0.0/0

(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

Port (Service)	Source	Action
ALL		
ALL	sg-e475988b	Delete
TCP		
22 (SSH)	0.0.0.0/0	Delete

2 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 ICMP traffic from anywhere.



	Name	VPC	Description
☑	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

1 Security Group selected

Security Group: default

Create a new rule: Custom TCP rule

Port range: (e.g., 80 or 49152-65535)

Source: 0.0.0.0/0 (e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

ALL		
Port (Service)	Source	Action
ALL	sg-e475988b	Delete
ICMP		
Port (Service)	Source	Action
ALL	0.0.0.0/0	Delete
TCP		
Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete

NOTE *If you wish 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.*

Obtaining and Launching the Vyatta AMI

This section presents the following topics:

- Obtaining the Vyatta AMI from the EC2 Console
- Obtaining the Vyatta AMI from the AWS Marketplace

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**.

It is supported as either a **Small** (**m1.small**), **Medium** (**m1.medium**), or **Large** (**m1.Large**) instance within AWS and is provided with persistent Amazon Elastic Block Storage (EBS).

There are two different ways to purchase the Vyatta AMI, depending on your business needs: as a term contract from Vyatta, and on a pay-as-you-go basis from the AWS Marketplace.

Purchasing a term contract from Vyatta is more cost-effective for longer-term usage. If you have purchased a term contract from Vyatta, you obtain the Vyatta AMI from the EC2 console. To do this, see "Obtaining the Vyatta AMI from the EC2 Console" on page 11.

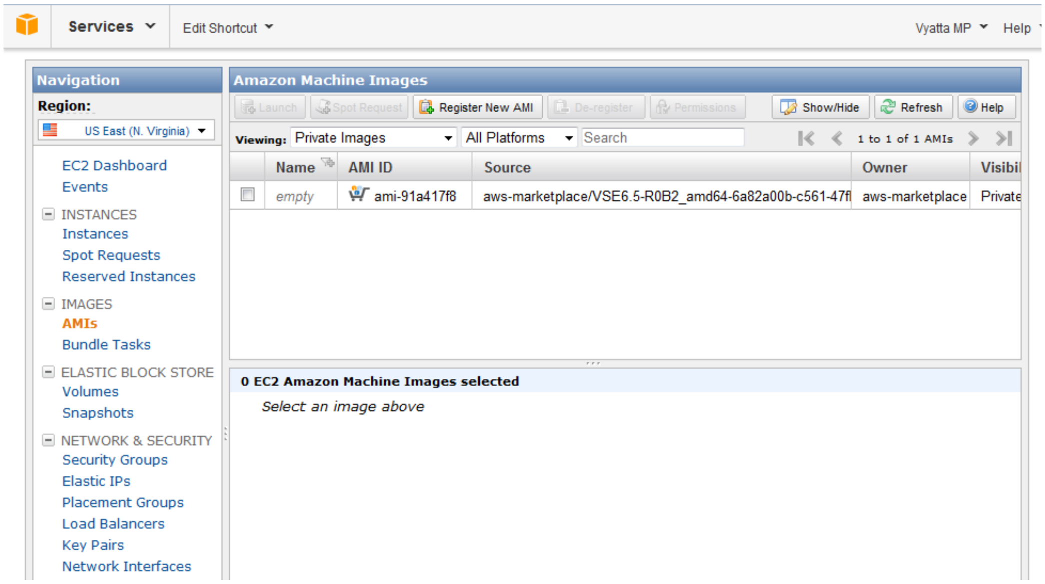
Accessing the Vyatta AMI on a pay-as-you-go basis from the AWS Marketplace provides a flexible alternative to a term contract. If you wish to use the Vyatta AMI on a pay-as-you-go basis, you obtain it from the AWS Marketplace. To do this, see "Obtaining the Vyatta AMI from the AWS Marketplace" on page 20.

Obtaining the Vyatta AMI from the EC2 Console

When you purchase a Vyatta AMI term contract, you must provide Vyatta with your AWS account number so that Vyatta can share the Vyatta AMI with you in the AWS environment.

To obtain and launch the Vyatta AMI from the EC2 Console

1. Click **EC2** on the AWS Management Console Home page. The **Amazon EC2 Console Dashboard** page appears.
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.

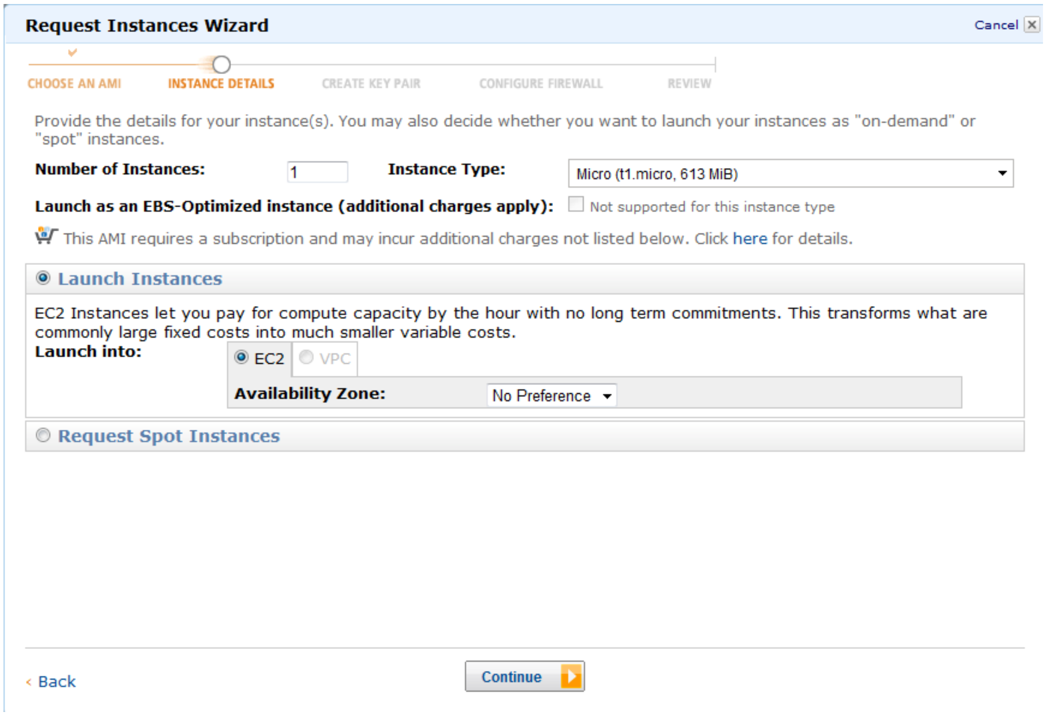


	Name	AMI ID	Source	Owner	Visibil
☐	*empty*	ami-91a417f8	aws-marketplace/VSE6.5-R0B2_amd64-6a82a00b-c561-47f	aws-marketplace	Private

0 EC2 Amazon Machine Images selected

Select an image above

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.



Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** Micro (t1.micro, 613 MiB)

Launch as an EBS-Optimized instance (additional charges apply): ☐ Not supported for this instance type

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

● **Launch Instances**

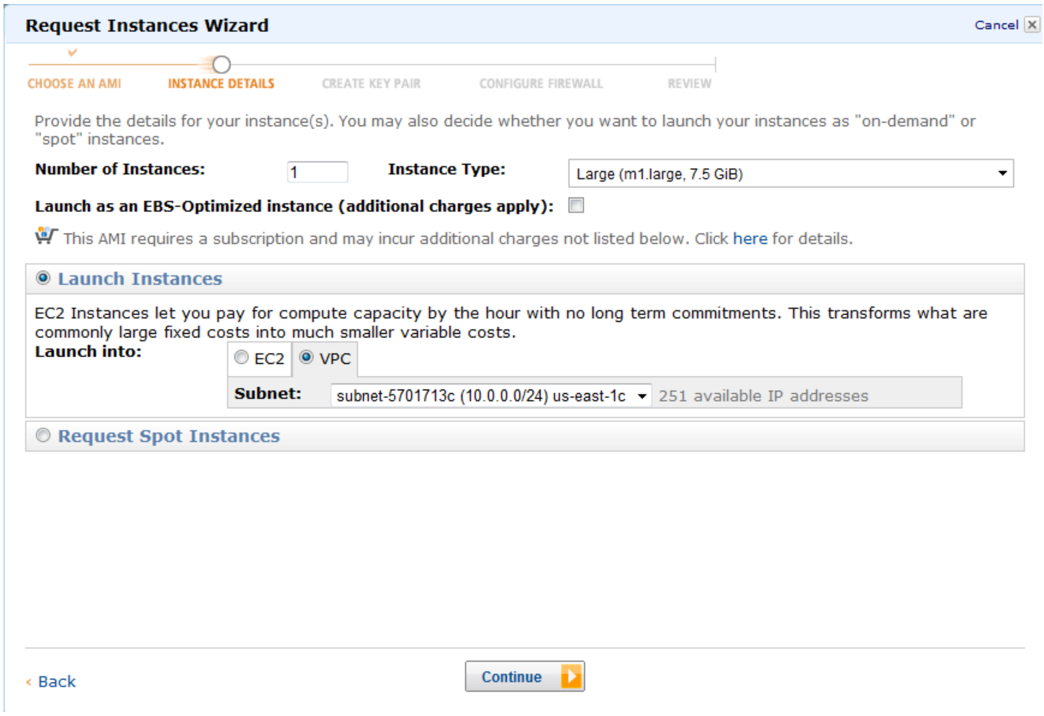
EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

Launch into: ● EC2 ○ VPC

Availability Zone: No Preference

○ **Request Spot Instances**

5 Select an **Instance Type** other than Micro in order to launch the Vyatta AMI instance into a VPC. Then, in the **Launch Instances** area, select **VPC** and select the subnet in the VPC that you wish to launch the instance into.



Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** Large (m1.large, 7.5 GiB)

Launch as an EBS-Optimized instance (additional charges apply): □

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

● **Launch Instances**

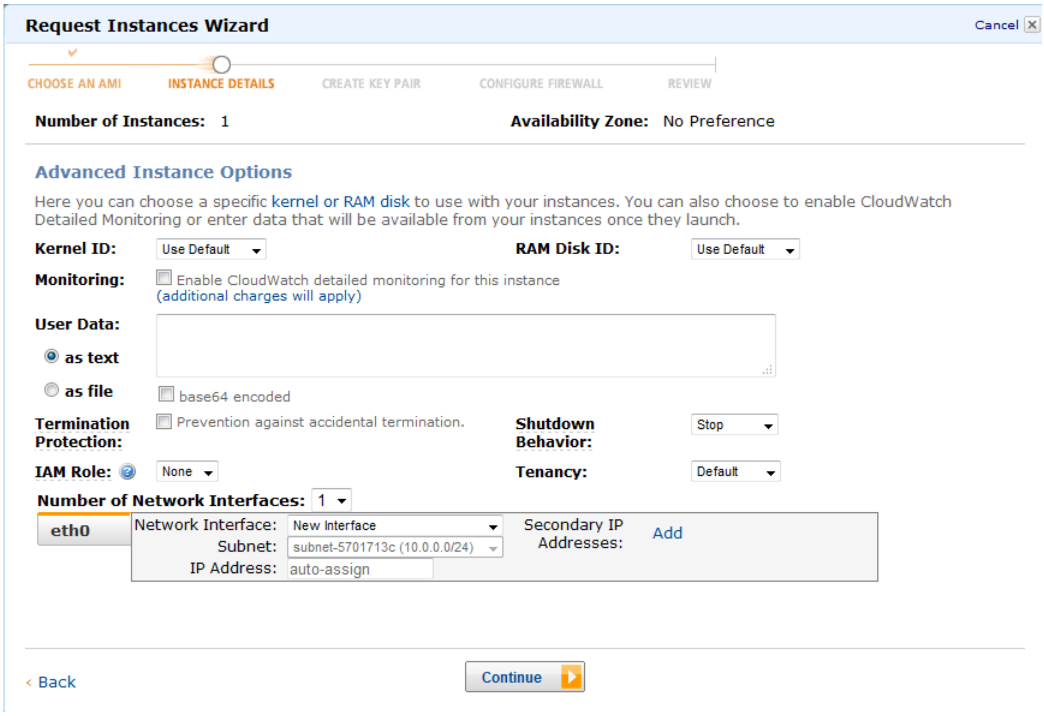
EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

Launch into: ○ EC2 ● VPC

Subnet: subnet-5701713c (10.0.0.0/24) us-east-1c 251 available IP addresses

○ **Request Spot Instances**

6 Click **Continue** to configure additional instance details.



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

Termination Protection: □ Prevention against accidental termination. **Shutdown Behavior:** Stop

IAM Role: None **Tenancy:** Default

Number of Network Interfaces: 1

eth0

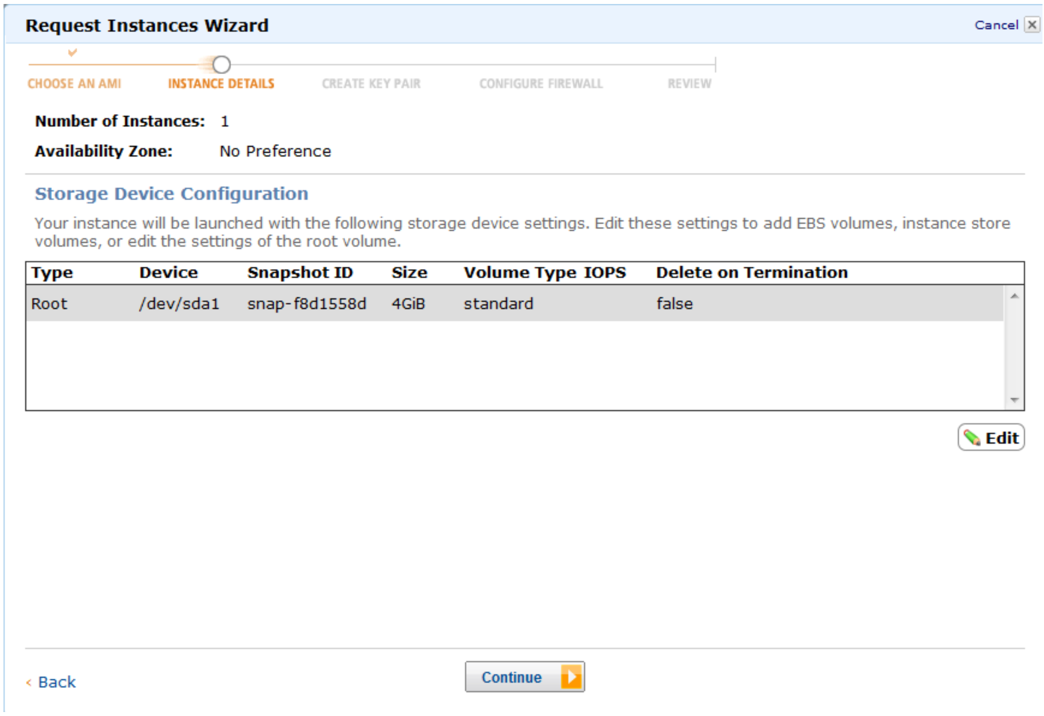
Network Interface: New Interface

Subnet: subnet-5701713c (10.0.0.0/24)

IP Address: auto-assign

Secondary IP Addresses: Add

7 If you wish to use a static IP address, specify the address in the **IP Address** field. If you wish to include more than one network interface, specify the number you wish to include in the **Number of Network Interfaces** field and configure them as required. Click **Continue** to configure additional instance details.



Number of Instances: 1

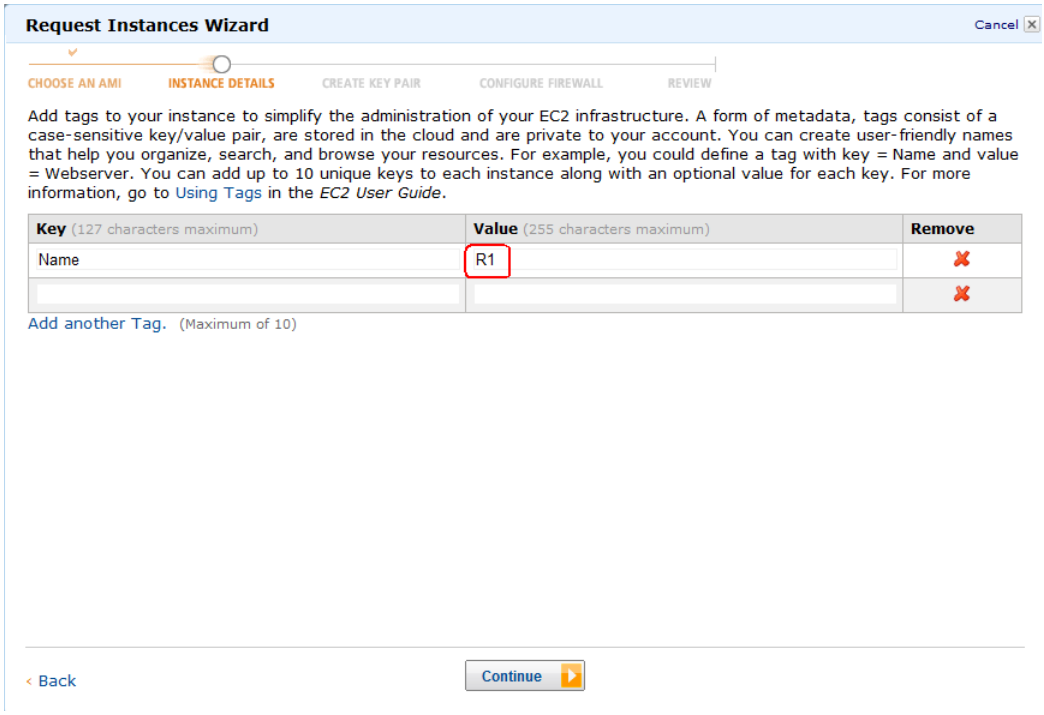
Availability Zone: No Preference

Storage Device Configuration

Your instance will be launched with the following storage device settings. Edit these settings to add EBS volumes, instance store volumes, or edit the settings of the root volume.

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination
Root	/dev/sda1	snap-f8d1558d	4GiB	standard		false

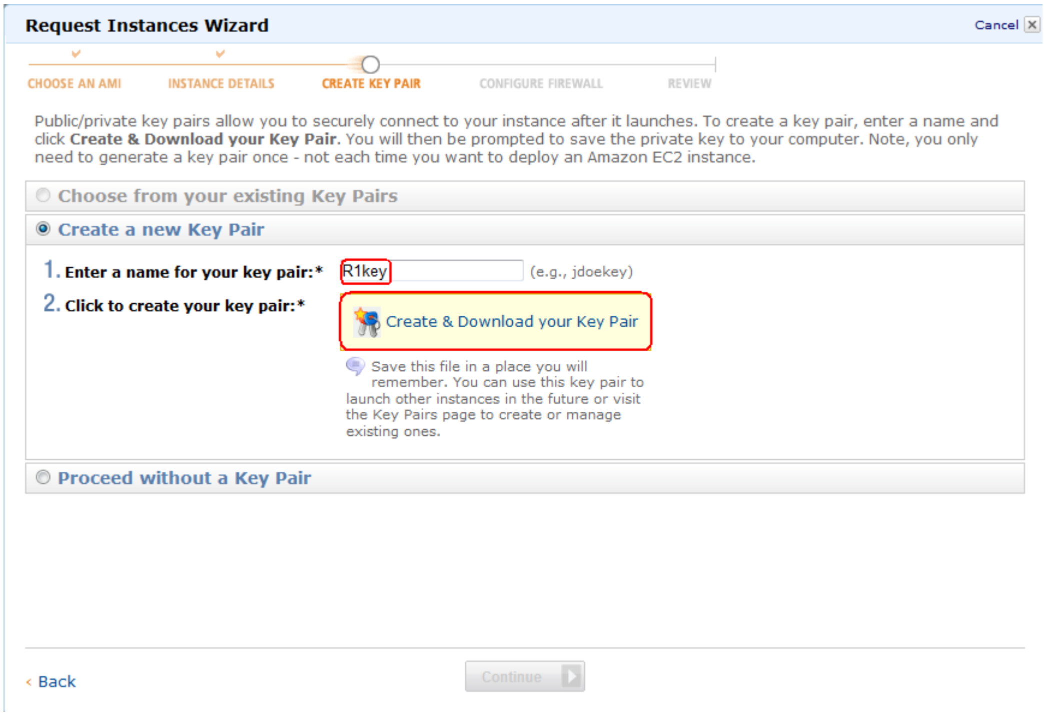
8 If you wish to modify the storage configuration, click the **Edit** button and configure it as required. Click **Continue** to configure additional instance details.



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	R1	✘
		✘

9 If you wish to add tags (for example, a name) to your instance, specify a **Key** and an associated **Value**. In this case we have chosen a name of R1. Click **Continue** to move to the **Create Key Pair** page.



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

● Create a new Key Pair

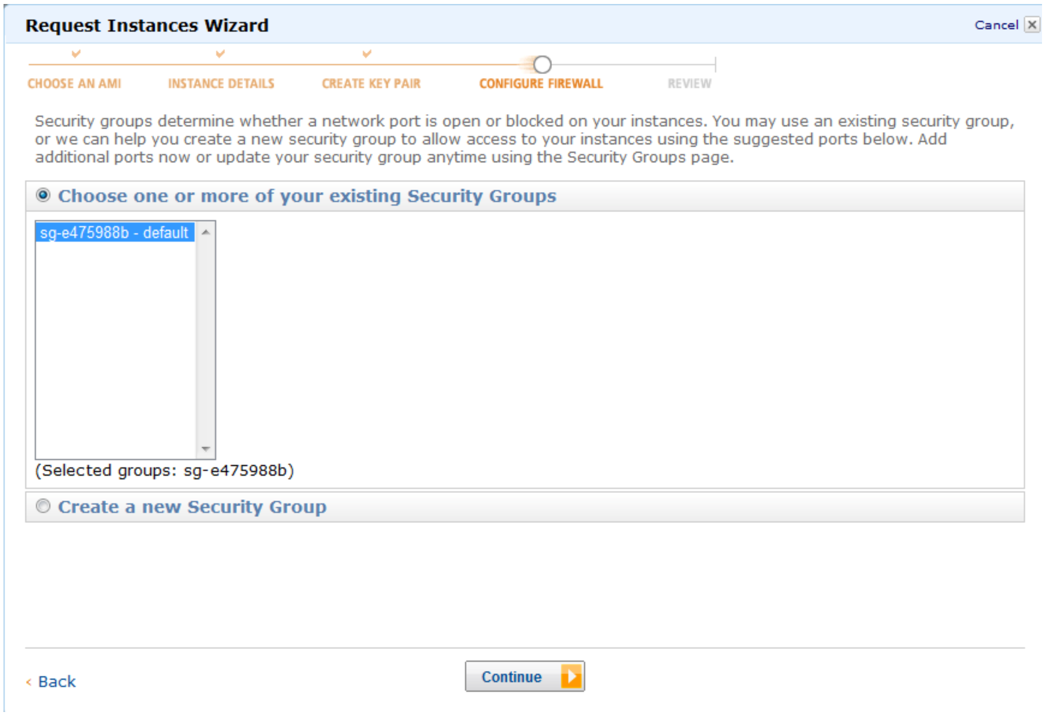
1. **Enter a name for your key pair:*** R1key (e.g., jdoekey)
2. **Click to create your key pair:***

Save this file in a place you will remember. You can use this key pair to launch other instances in the future or visit the Key Pairs page to create or manage existing ones.

○ Proceed without a Key Pair

10 At this point, you can select **Proceed without a Key Pair** if a secure connection to your Vyatta AMI instance is not required. For our example, we will create a key pair. Select **Create a new Key Pair.** Enter a name for the key pair in the **Enter a name for your key pair** field (in this case we entered “R1key”). Click the **Create**

& Download you Key Pair button. Save the .pem key pair file. You will move to the **Configure Firewall** page.



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.

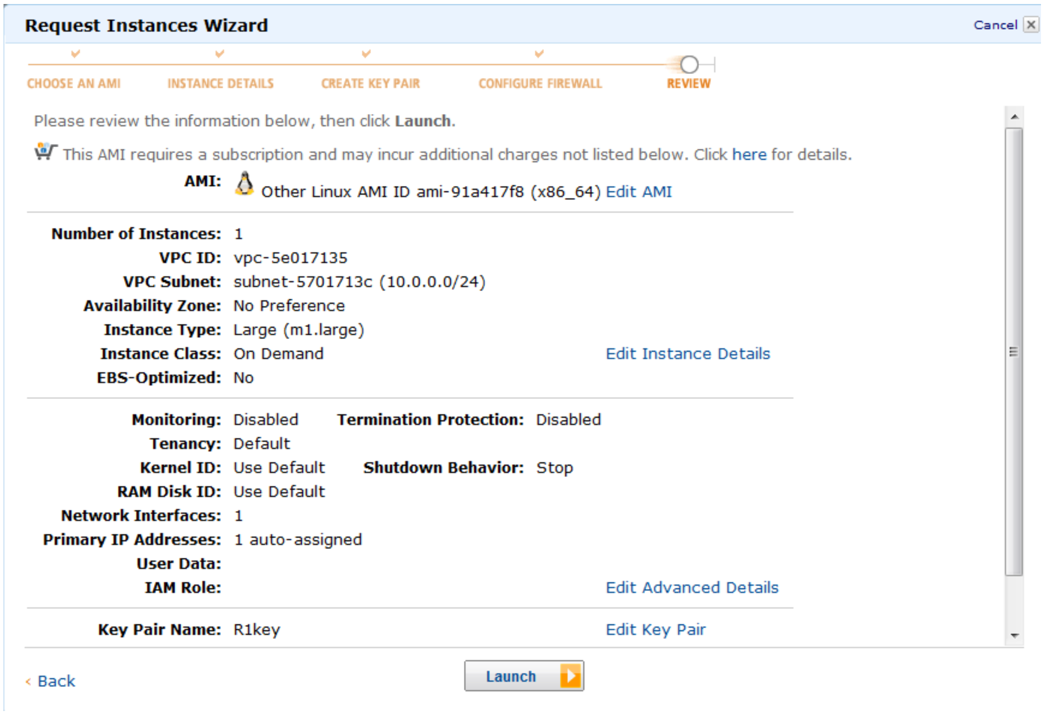
● Choose one or more of your existing Security Groups

sg-e475988b - default

(Selected groups: sg-e475988b)

○ Create a new Security Group

11 In the **Choose one or more of your existing Security Groups** area, select a security group that you created earlier. Click **Continue** to move to the **Review** page.



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

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

AMI: Other Linux AMI ID ami-91a417f8 (x86_64) Edit AMI

Number of Instances: 1
VPC ID: vpc-5e017135
VPC Subnet: subnet-5701713c (10.0.0.0/24)
Availability Zone: No Preference
Instance Type: Large (m1.large)
Instance Class: On Demand
EBS-Optimized: No

Edit Instance Details

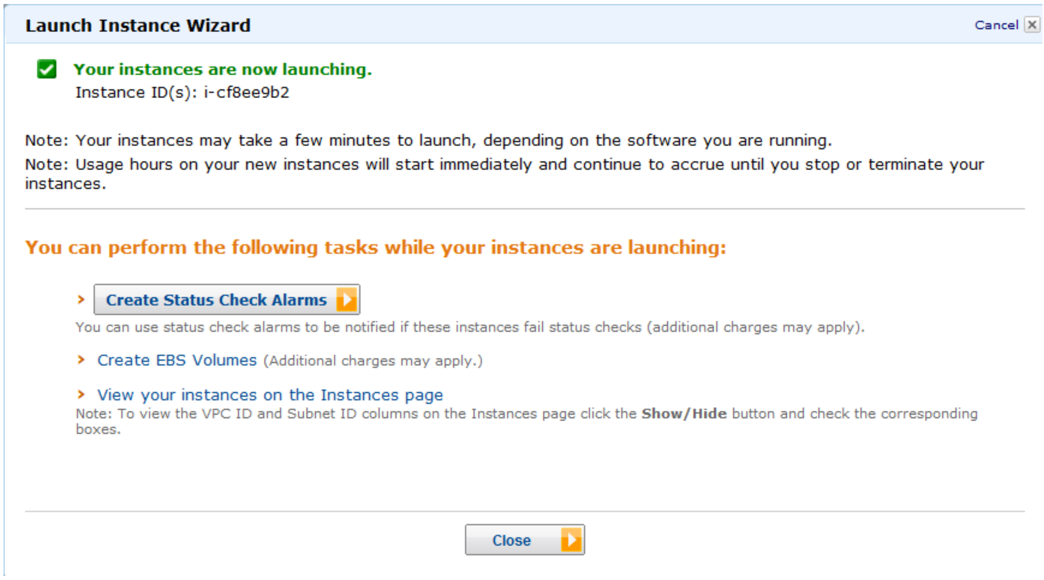
Monitoring: Disabled **Termination Protection:** Disabled
Tenancy: Default
Kernel ID: Use Default **Shutdown Behavior:** Stop
RAM Disk ID: Use Default
Network Interfaces: 1
Primary IP Addresses: 1 auto-assigned
User Data:
IAM Role:

Edit Advanced Details

Key Pair Name: R1key

Edit Key Pair

12 Review that the information that you have configured is correct. Click the **Launch** button to launch the Vyatta AMI instance into your VPC. The **Launch Instance Wizard** page appears.



Launch Instance Wizard

☑ Your instances are now launching.
Instance ID(s): i-cf8ee9b2

Note: Your instances may take a few minutes to launch, depending on the software you are running.

Note: Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

You can perform the following tasks while your instances are launching:

› Create Status Check Alarms
You can use status check alarms to be notified if these instances fail status checks (additional charges may apply).

› Create EBS Volumes (Additional charges may apply.)

› View your instances on the Instances page
Note: To view the VPC ID and Subnet ID columns on the Instances page click the **Show/Hide** button and check the corresponding boxes.

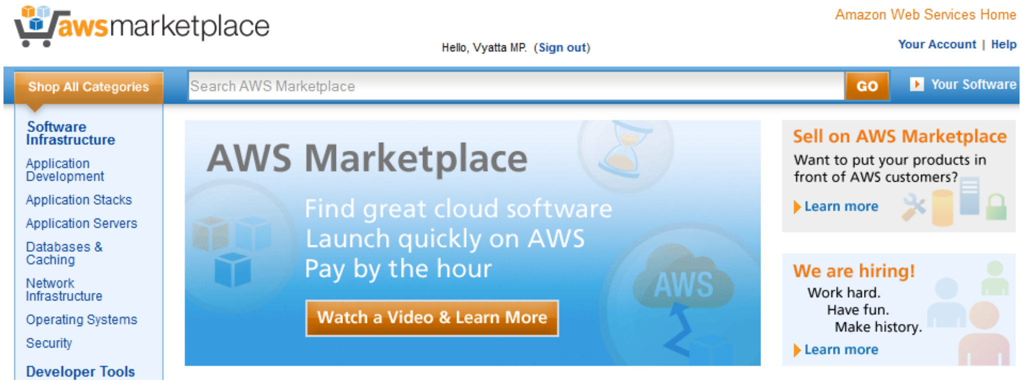
13 Click Close to return to the Amazon EC2 Console. At this point the Vyatta AMI instance is running within your VPC. The next step is to assign an Elastic IP address to the Vyatta AMI instance. See “Assigning an AWS Elastic IP Address to the Instance” on page 33.

Obtaining the Vyatta AMI from the AWS Marketplace

If you intend to use the Vyatta AMI on a pay-as-you-go basis, you must obtain it from the Amazon AWS Marketplace. The AWS Marketplace can be accessed once you are logged on to AWS.

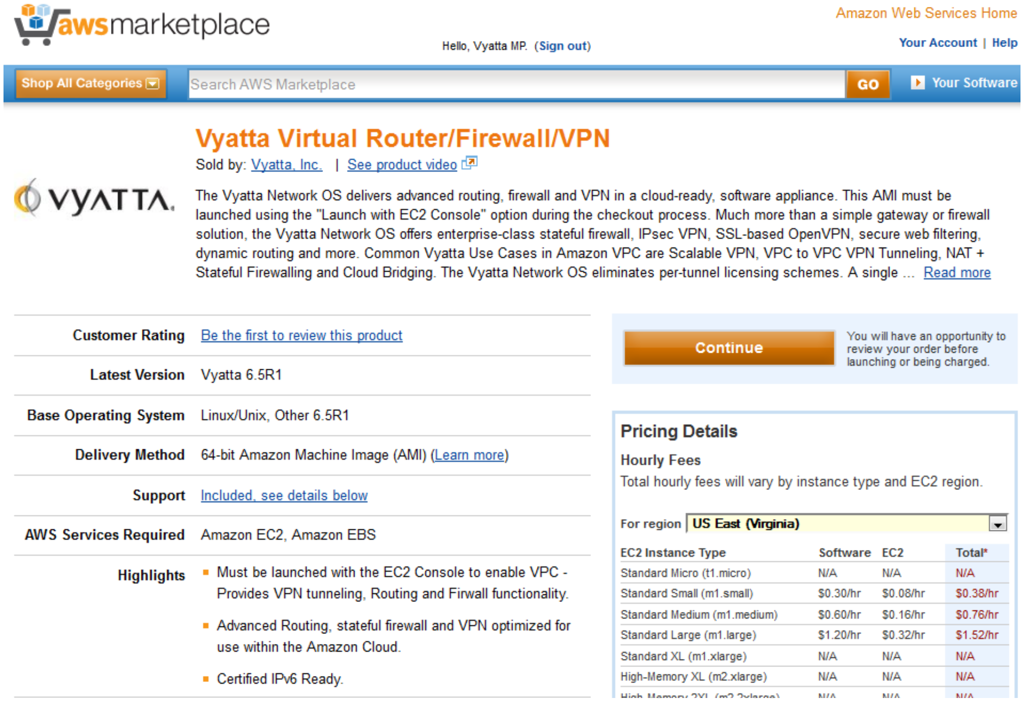
To obtain and launch the Vyatta AMI from the AWS Marketplace

1 Go to the AWS Marketplace at https://aws.amazon.com/marketplace/ and click **Sign on** at the top of the page. The AWS Marketplace main page appears.



2 In the **Search AWS Marketplace** field at the top of the page, enter **Vyatta** and click **GO**.

3 Select the 64-bit Vyatta AMI for the current Vyatta release. The Vyatta Virtual Router/Firewall/VPN page appears.



Vyatta Virtual Router/Firewall/VPN

Sold by: Vyatta, Inc. | See product video

The Vyatta Network OS delivers advanced routing, firewall and VPN in a cloud-ready, software appliance. This AMI must be launched using the "Launch with EC2 Console" option during the checkout process. Much more than a simple gateway or firewall solution, the Vyatta Network OS offers enterprise-class stateful firewall, IPsec VPN, SSL-based OpenVPN, secure web filtering, dynamic routing and more. Common Vyatta Use Cases in Amazon VPC are Scalable VPN, VPC to VPC VPN Tunneling, NAT + Stateful Firewalling and Cloud Bridging. The Vyatta Network OS eliminates per-tunnel licensing schemes. A single ... Read more

Customer Rating	Be the first to review this product
Latest Version	Vyatta 6.5R1
Base Operating System	Linux/Unix, Other 6.5R1
Delivery Method	64-bit Amazon Machine Image (AMI) (Learn more)
Support	Included, see details below
AWS Services Required	Amazon EC2, Amazon EBS
Highlights	▪ Must be launched with the EC2 Console to enable VPC - Provides VPN tunneling, Routing and Firwall functionality. ▪ Advanced Routing, stateful firewall and VPN optimized for use within the Amazon Cloud. ▪ Certified IPv6 Ready.

You will have an opportunity to review your order before launching or being charged.

Pricing Details

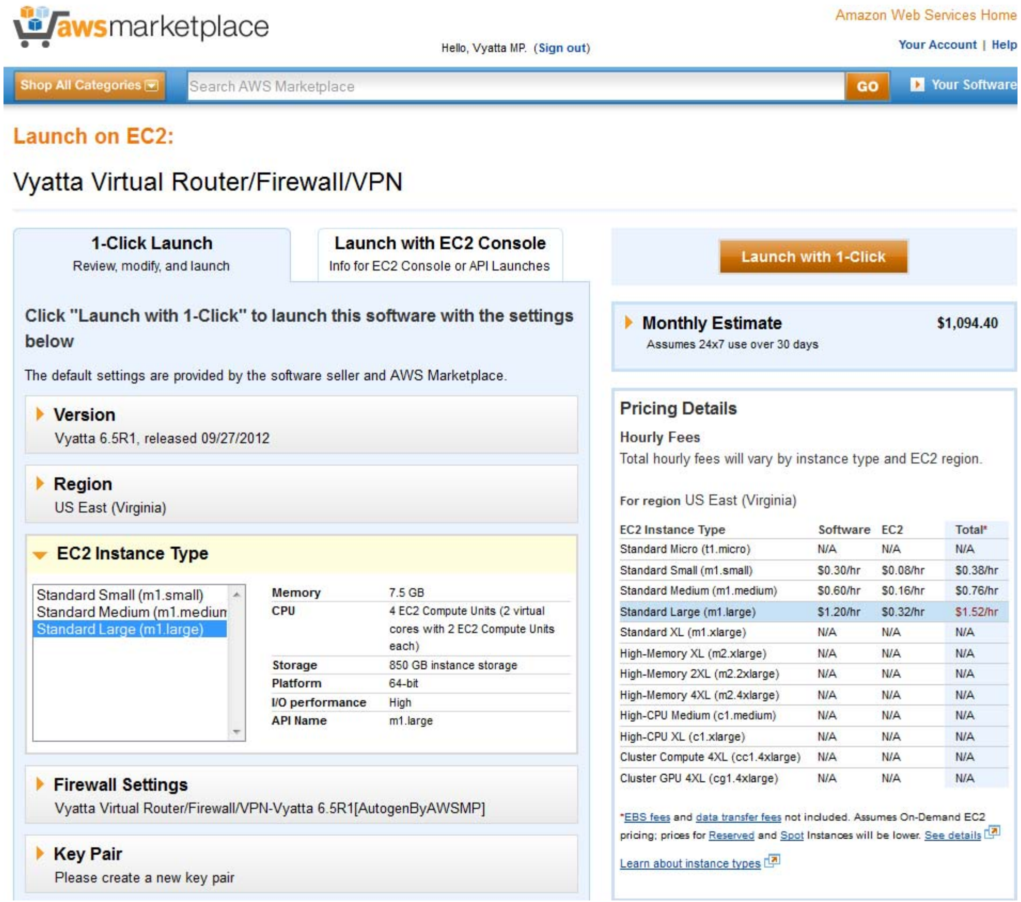
Hourly Fees

Total hourly fees will vary by instance type and EC2 region.

For region: US East (Virginia)

EC2 Instance Type	Software	EC2	Total*
Standard Micro (t1.micro)	N/A	N/A	N/A
Standard Small (m1.small)	$0.30/hr	$0.08/hr	$0.38/hr
Standard Medium (m1.medium)	$0.60/hr	$0.16/hr	$0.76/hr
Standard Large (m1.large)	$1.20/hr	$0.32/hr	$1.52/hr
Standard XL (m1.xlarge)	N/A	N/A	N/A
High-Memory XL (m2.xlarge)	N/A	N/A	N/A
High-Memory 2XL (m2.2xlarge)	N/A	N/A	N/A

4 Click Continue. The **Launch on EC2: Vyatta Virtual Router/Firewall/VPN** page appears.



Launch on EC2:

Vyatta Virtual Router/Firewall/VPN

Click "Launch with 1-Click" to launch this software with the settings below

The default settings are provided by the software seller and AWS Marketplace.

Version
Vyatta 6.5R1, released 09/27/2012

Region
US East (Virginia)

EC2 Instance Type

Standard Small (m1.small)
Standard Medium (m1.mediun
Standard Large (m1.large)

Memory	7.5 GB
CPU	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)
Storage	850 GB instance storage
Platform	64-bit
I/O performance	High
API Name	m1.large

Firewall Settings
Vyatta Virtual Router/Firewall/VPN-Vyatta 6.5R1[AutogenByAWSMP]

Key Pair
Please create a new key pair

Monthly Estimate $1,094.40
Assumes 24x7 use over 30 days

Pricing Details

Hourly Fees
Total hourly fees will vary by instance type and EC2 region.

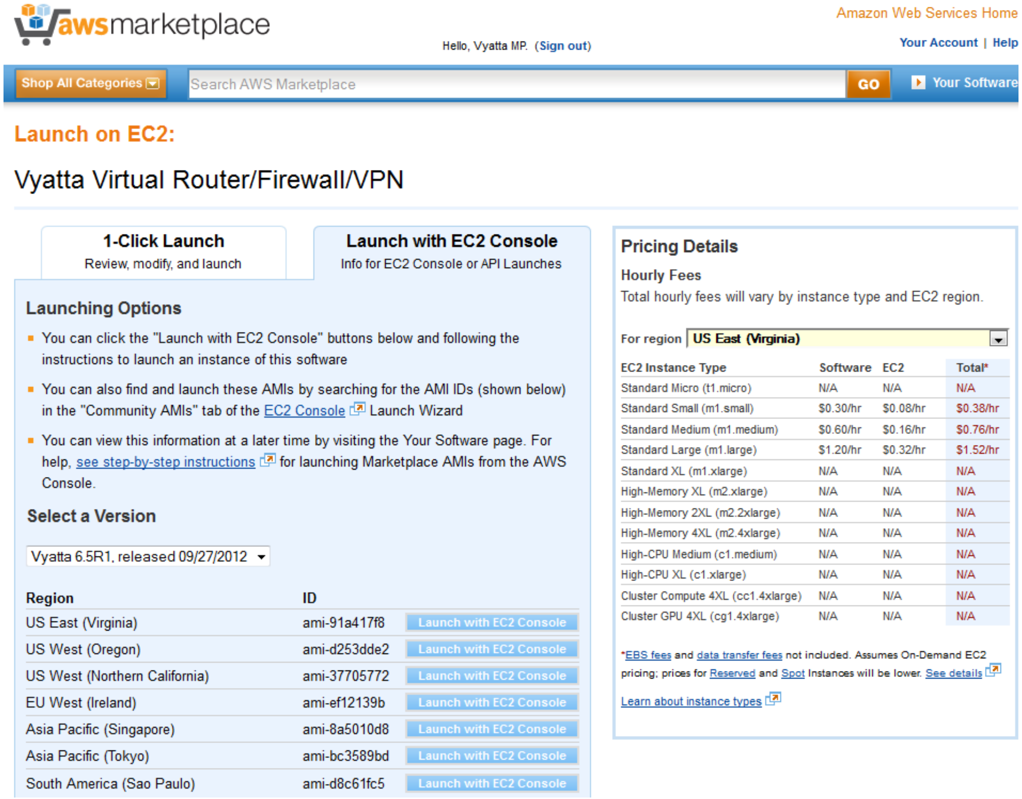
For region US East (Virginia)

EC2 Instance Type	Software	EC2	Total*
Standard Micro (t1.micro)	N/A	N/A	N/A
Standard Small (m1.small)	$0.30/hr	$0.08/hr	$0.38/hr
Standard Medium (m1.medium)	$0.60/hr	$0.16/hr	$0.76/hr
Standard Large (m1.large)	$1.20/hr	$0.32/hr	$1.52/hr
Standard XL (m1.xlarge)	N/A	N/A	N/A
High-Memory XL (m2.xlarge)	N/A	N/A	N/A
High-Memory 2XL (m2.2xlarge)	N/A	N/A	N/A
High-Memory 4XL (m2.4xlarge)	N/A	N/A	N/A
High-CPU Medium (c1.medium)	N/A	N/A	N/A
High-CPU XL (c1.xlarge)	N/A	N/A	N/A
Cluster Compute 4XL (cc1.4xlarge)	N/A	N/A	N/A
Cluster GPU 4XL (cg1.4xlarge)	N/A	N/A	N/A

*EBS fees and data transfer fees not included. Assumes On-Demand EC2 pricing; prices for Reserved and Spot Instances will be lower. See details

Learn about instance types

5 At this point, if you wish to simply create an instance of the Vyatta AMI within the EC2 infrastructure, you can click **Launch with 1-Click** and follow the prompts. You can then access the Vyatta AMI instance using an SSH client and specifying the assigned DNS name. For the examples that follow, we want to launch the instance into a VPC. To do this, select the **Launch with EC2 Console** tab.



Launch on EC2:

Vyatta Virtual Router/Firewall/VPN

Launching Options

- You can click the "Launch with EC2 Console" buttons below and following the instructions to launch an instance of this software
- You can also find and launch these AMIs by searching for the AMI IDs (shown below) in the "Community AMIs" tab of the EC2 Console Launch Wizard
- You can view this information at a later time by visiting the Your Software page. For help, see step-by-step instructions for launching Marketplace AMIs from the AWS Console.

Select a Version

Vyatta 6.5R1, released 09/27/2012

Region	ID	
US East (Virginia)	ami-91a417f8	Launch with EC2 Console
US West (Oregon)	ami-d253dde2	Launch with EC2 Console
US West (Northern California)	ami-37705772	Launch with EC2 Console
EU West (Ireland)	ami-ef12139b	Launch with EC2 Console
Asia Pacific (Singapore)	ami-8a5010d8	Launch with EC2 Console
Asia Pacific (Tokyo)	ami-bc3589bd	Launch with EC2 Console
South America (Sao Paulo)	ami-d8c61fc5	Launch with EC2 Console

Pricing Details

Hourly Fees

Total hourly fees will vary by instance type and EC2 region.

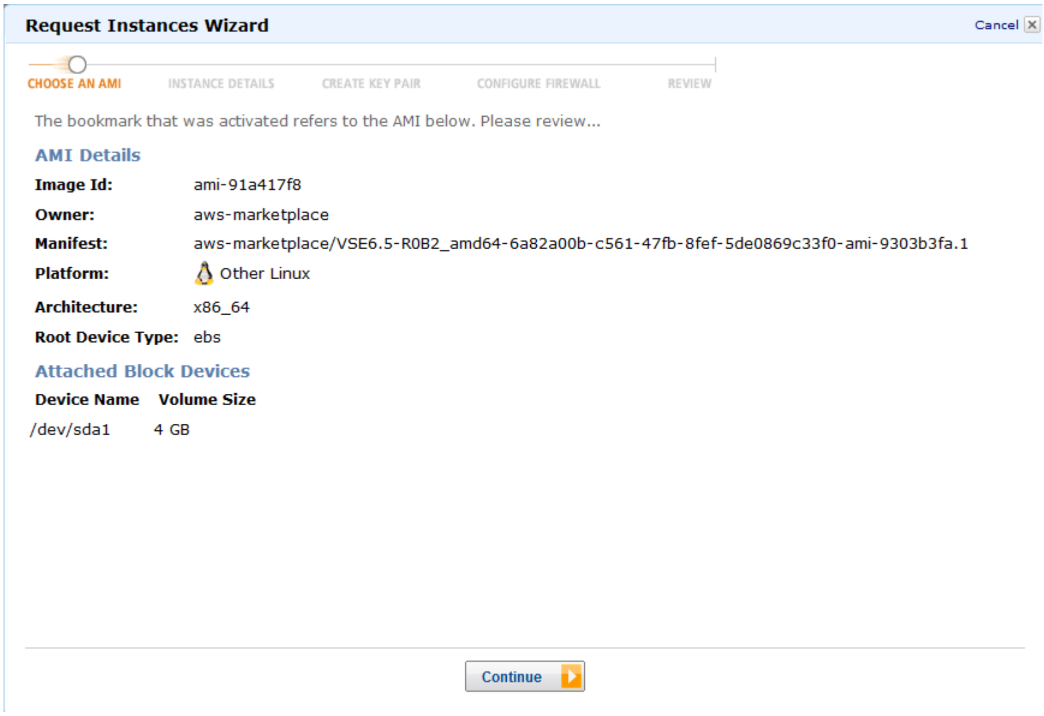
For region: US East (Virginia)

EC2 Instance Type	Software	EC2	Total*
Standard Micro (t1.micro)	N/A	N/A	N/A
Standard Small (m1.small)	$0.30/hr	$0.08/hr	$0.38/hr
Standard Medium (m1.medium)	$0.60/hr	$0.16/hr	$0.76/hr
Standard Large (m1.large)	$1.20/hr	$0.32/hr	$1.52/hr
Standard XL (m1.xlarge)	N/A	N/A	N/A
High-Memory XL (m2.xlarge)	N/A	N/A	N/A
High-Memory 2XL (m2.2xlarge)	N/A	N/A	N/A
High-Memory 4XL (m2.4xlarge)	N/A	N/A	N/A
High-CPU Medium (c1.medium)	N/A	N/A	N/A
High-CPU XL (c1.xlarge)	N/A	N/A	N/A
Cluster Compute 4XL (cc1.4xlarge)	N/A	N/A	N/A
Cluster GPU 4XL (cg1.4xlarge)	N/A	N/A	N/A

*EBS fees and data transfer fees not included. Assumes On-Demand EC2 pricing; prices for Reserved and Spot Instances will be lower. See details

Learn about instance types

6 Click the **Launch with EC2 Console** button next to the AWS region you wish to launch it in. The **Request Instances Wizard** appears at the **Choose and AMI** page in the EC2 console.



The bookmark that was activated refers to the AMI below. Please review...

AMI Details

Image Id: ami-91a417f8

Owner: aws-marketplace

Manifest: aws-marketplace/VSE6.5-R0B2_amd64-6a82a00b-c561-47fb-8fef-5de0869c33f0-ami-9303b3fa.1

Platform: Other Linux

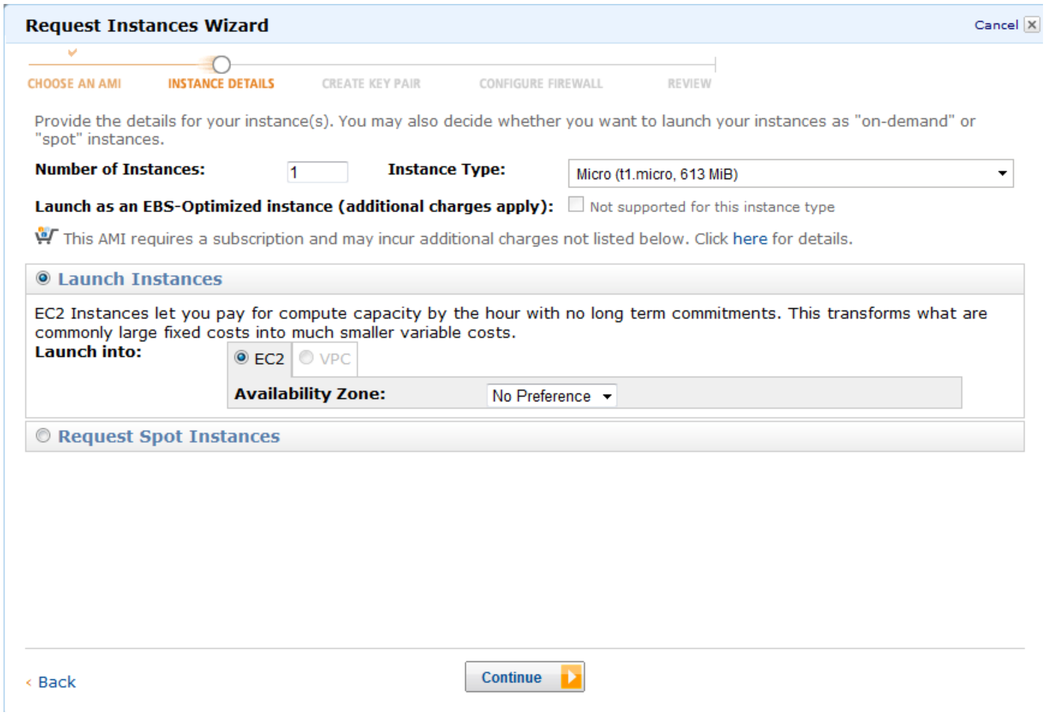
Architecture: x86_64

Root Device Type: ebs

Attached Block Devices

Device Name	Volume Size
/dev/sda1	4 GB

7 Click **Continue** to move to the **Install Details** page.



Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** Micro (t1.micro, 613 MiB)

Launch as an EBS-Optimized instance (additional charges apply): □ Not supported for this instance type

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

● **Launch Instances**

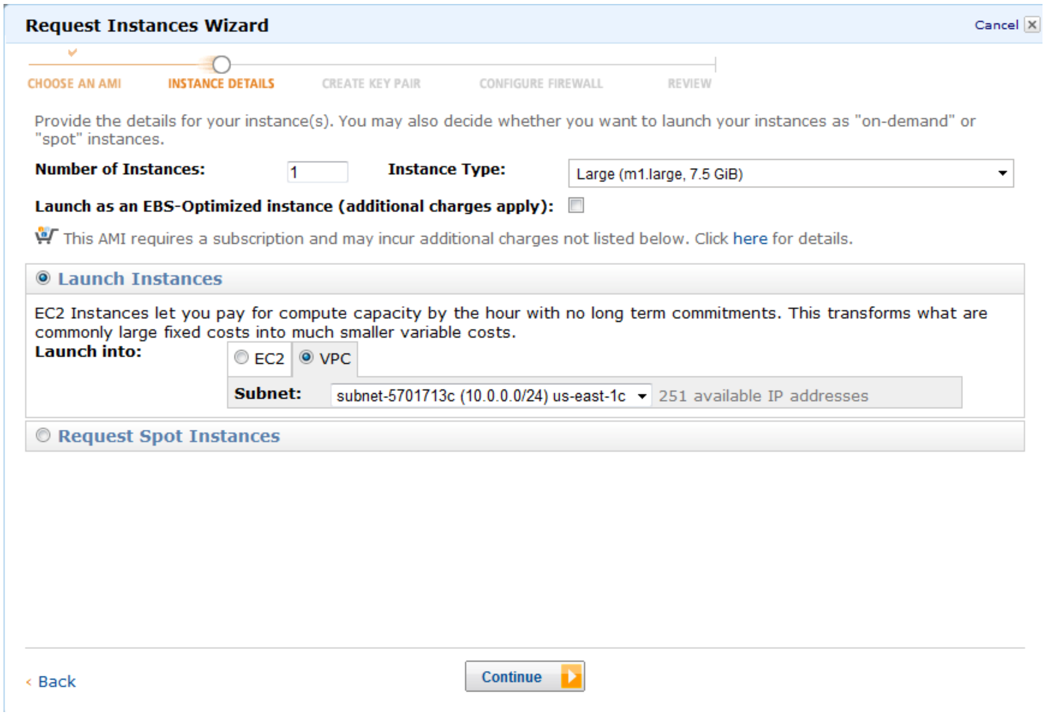
EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

Launch into: ● EC2 ○ VPC

Availability Zone: No Preference

○ **Request Spot Instances**

8 Select an **Instance Type** other than Micro in order to launch the Vyatta AMI instance into a VPC. Then, in the **Launch Instances** area, select **VPC** and select the subnet in the VPC that you wish to launch the instance into.



Request Instances Wizard

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** Large (m1.large, 7.5 GiB)

Launch as an EBS-Optimized instance (additional charges apply): □

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

● **Launch Instances**

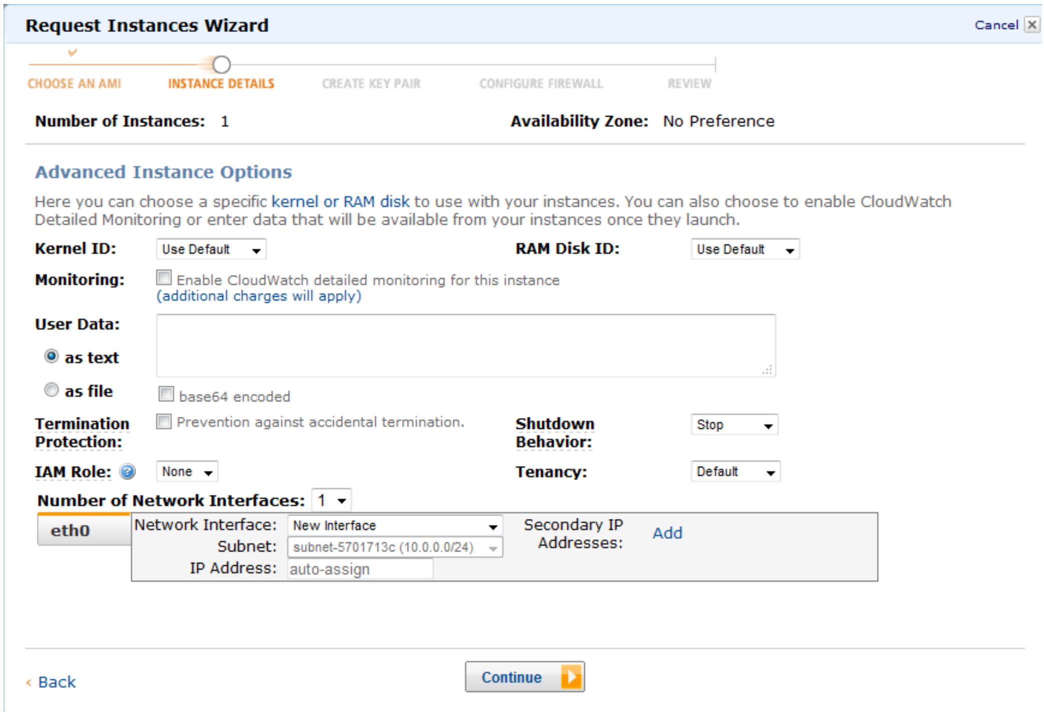
EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

Launch into: ○ EC2 ● VPC

Subnet: subnet-5701713c (10.0.0.0/24) us-east-1c 251 available IP addresses

○ **Request Spot Instances**

9 Click **Continue** to configure additional instance details.



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

Termination Protection: □ Prevention against accidental termination.

Shutdown Behavior: Stop

IAM Role: None

Tenancy: Default

Number of Network Interfaces: 1

eth0

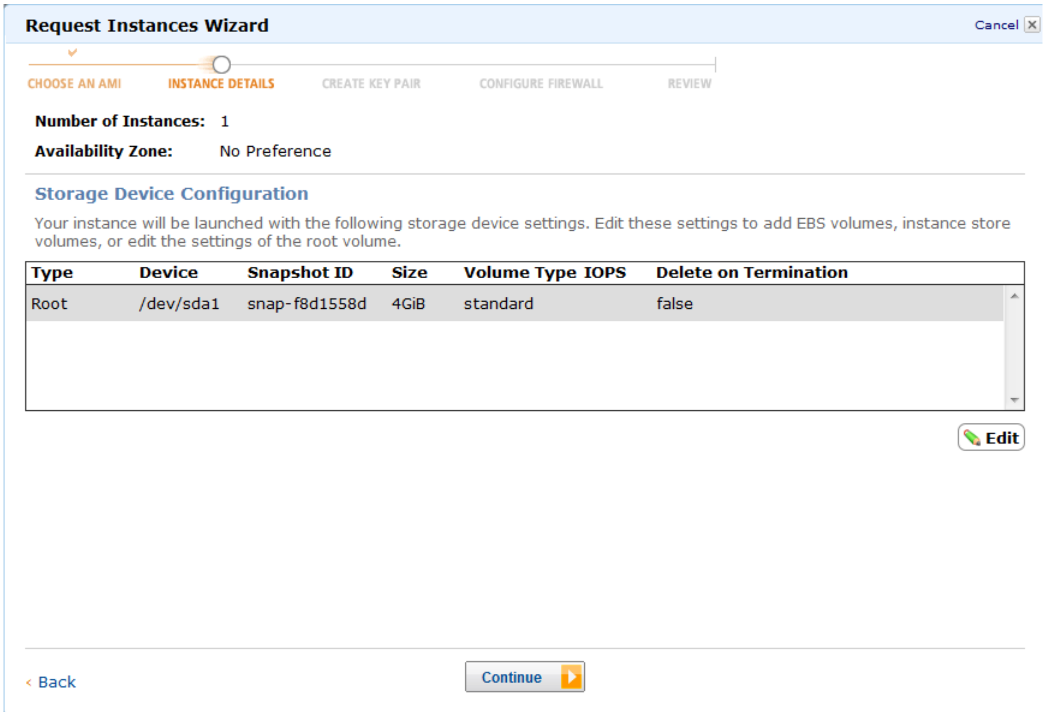
Network Interface: New Interface

Subnet: subnet-5701713c (10.0.0.0/24)

IP Address: auto-assign

Secondary IP Addresses: Add

10 If you wish to use a static IP address, specify the address in the **IP Address** field. If you wish to include more than one network interface, specify the number you wish to include in the **Number of Network Interfaces** field and configure them as required. Click **Continue** to configure additional instance details.



Number of Instances: 1

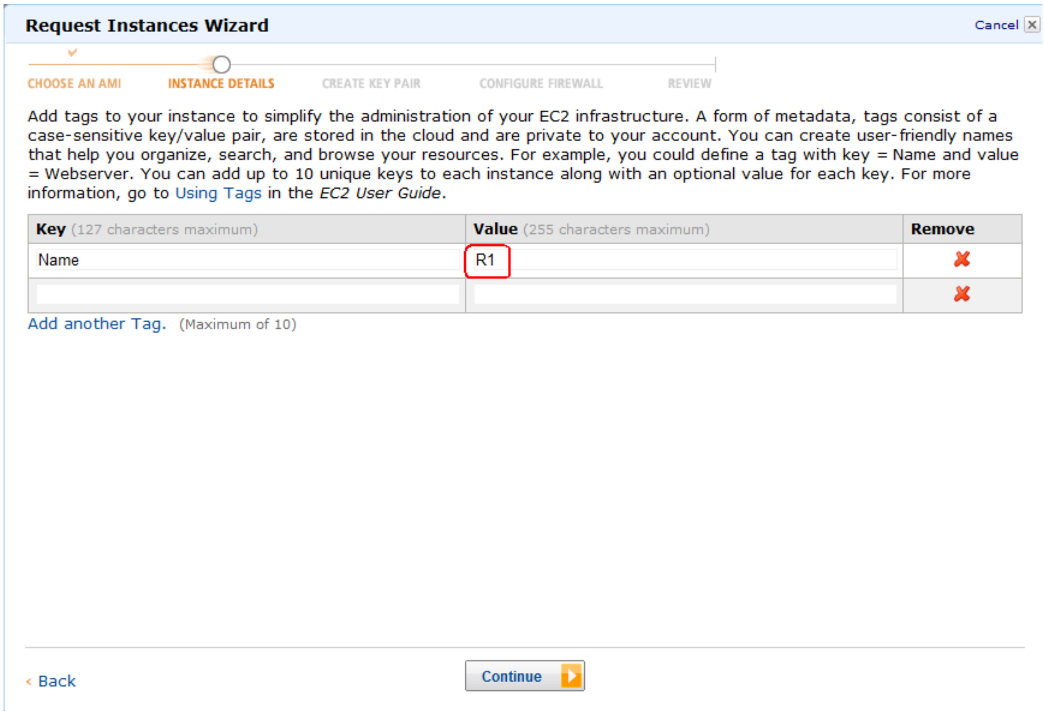
Availability Zone: No Preference

Storage Device Configuration

Your instance will be launched with the following storage device settings. Edit these settings to add EBS volumes, instance store volumes, or edit the settings of the root volume.

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination
Root	/dev/sda1	snap-f8d1558d	4GiB	standard		false

11 If you wish to modify the storage configuration, click the **Edit** button and configure it as required. Click **Continue** to configure additional instance details.

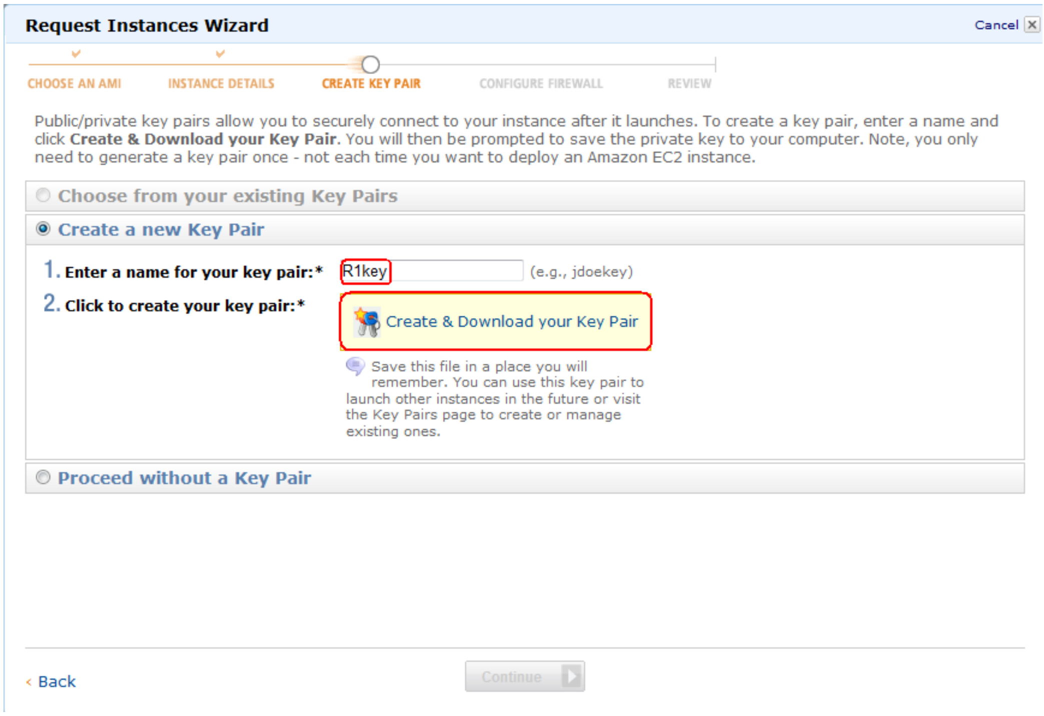


Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. 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	R1	✘
		✘

Add another Tag. (Maximum of 10)

12 If you wish to add tags (for example, a name) to your instance, specify a **Key** and an associated **Value**. In this case we have chosen a name of R1. Click **Continue** to move to the **Create Key Pair** page.



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

● Create a new Key Pair

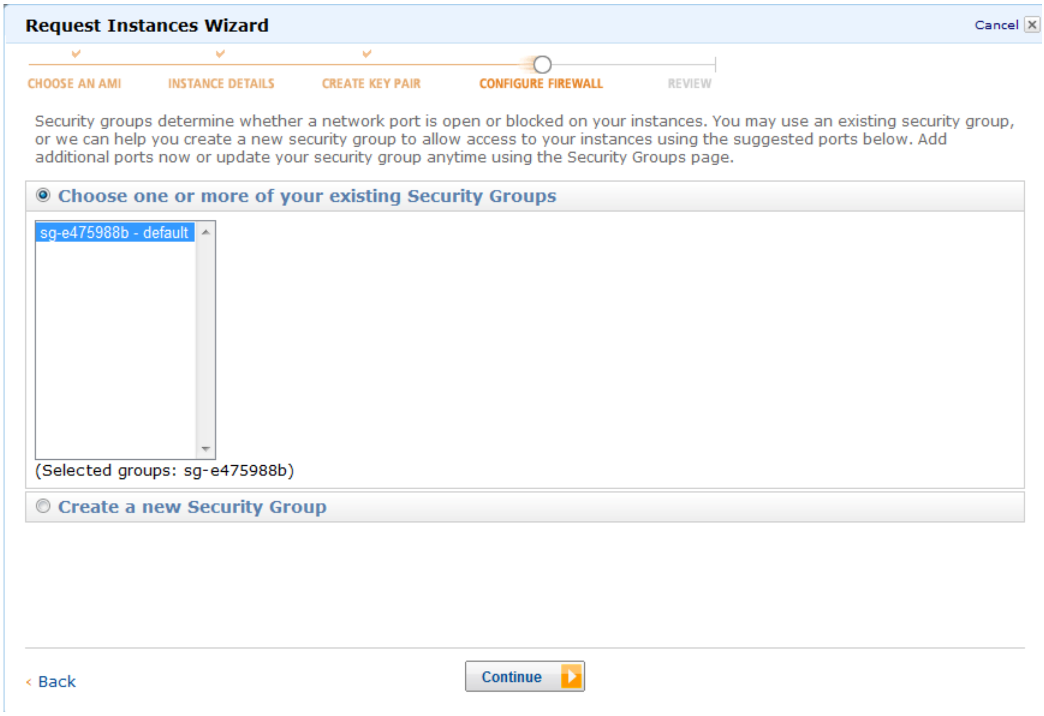
1. **Enter a name for your key pair:*** R1key (e.g., jdoekey)
2. **Click to create your key pair:*** Create & Download your Key Pair

Save this file in a place you will remember. You can use this key pair to launch other instances in the future or visit the Key Pairs page to create or manage existing ones.

○ Proceed without a Key Pair

13 At this point, you can select **Proceed without a Key Pair** if a secure connection to your Vyatta AMI instance is not required. For our example, we will create a key pair. Select **Create a new Key Pair**. Enter a name for the key pair in the **Enter a name for your key pair** field (in this case we entered “R1key”). Click the **Create**

& Download you Key Pair button. Save the .pem key pair file. You will move to the **Configure Firewall** page.



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.

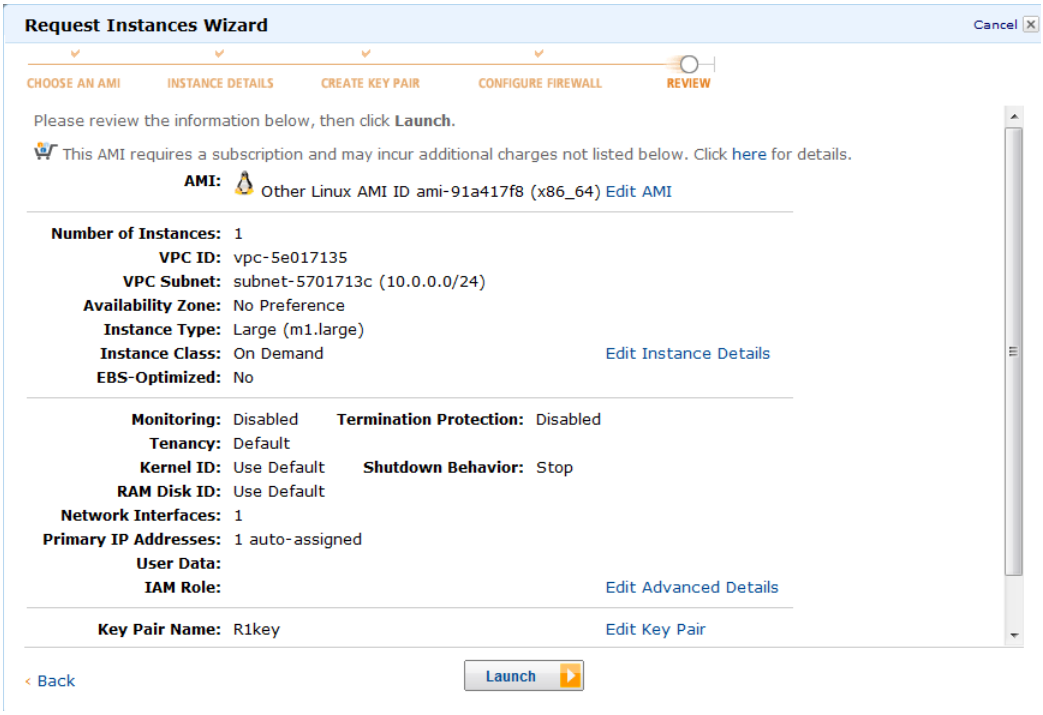
● Choose one or more of your existing Security Groups

sg-e475988b - default

(Selected groups: sg-e475988b)

○ Create a new Security Group

14 In the **Choose one or more of your existing Security Groups** area, select a security group that you created earlier. Click **Continue** to move to the **Review** page.



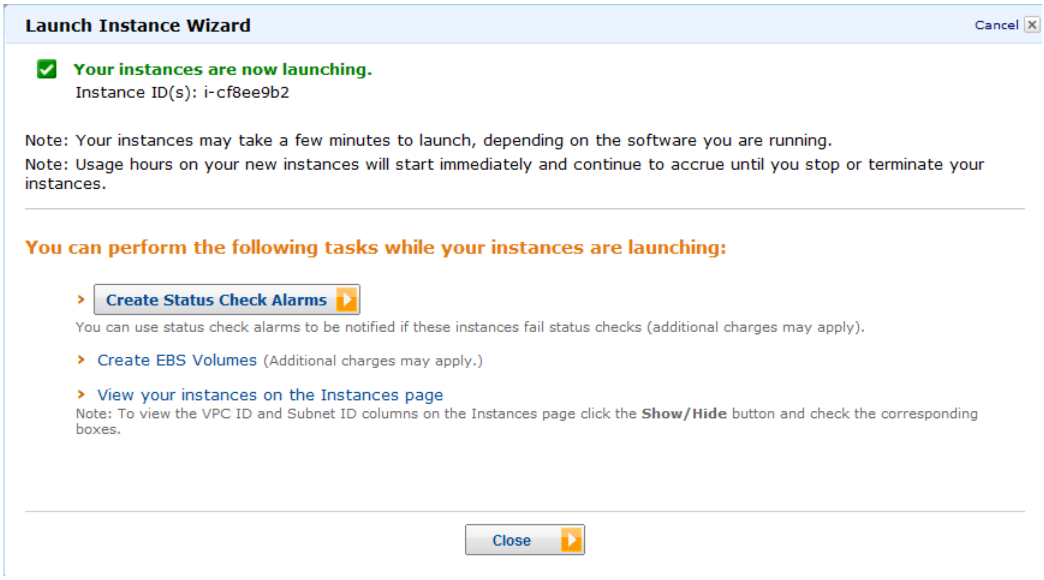
Request Instances Wizard

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

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

AMI:	Other Linux AMI ID ami-91a417f8 (x86_64) Edit AMI
Number of Instances:	1
VPC ID:	vpc-5e017135
VPC Subnet:	subnet-5701713c (10.0.0.0/24)
Availability Zone:	No Preference
Instance Type:	Large (m1.large)
Instance Class:	On Demand
EBS-Optimized:	No
Monitoring:	Disabled
Termination Protection:	Disabled
Tenancy:	Default
Kernel ID:	Use Default
Shutdown Behavior:	Stop
RAM Disk ID:	Use Default
Network Interfaces:	1
Primary IP Addresses:	1 auto-assigned
User Data:	
IAM Role:	
Key Pair Name:	R1key

15 Review that the information that you have configured is correct. Click the **Launch** button to launch the Vyatta AMI instance into your VPC. The **Launch Instance Wizard** page appears.



☑ Your instances are now launching.
Instance ID(s): i-cf8ee9b2

Note: Your instances may take a few minutes to launch, depending on the software you are running.
Note: Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

You can perform the following tasks while your instances are launching:

You can use status check alarms to be notified if these instances fail status checks (additional charges may apply).

(Additional charges may apply.)

Note: To view the VPC ID and Subnet ID columns on the Instances page click the **Show/Hide** button and check the corresponding boxes.

16 Click Close to return to the Amazon EC2 Console. At this point the Vyatta AMI instance is running within your VPC. The next step is to assign an Elastic IP address to the Vyatta AMI instance. See "Assigning an AWS Elastic IP Address to the Instance" on page 33.

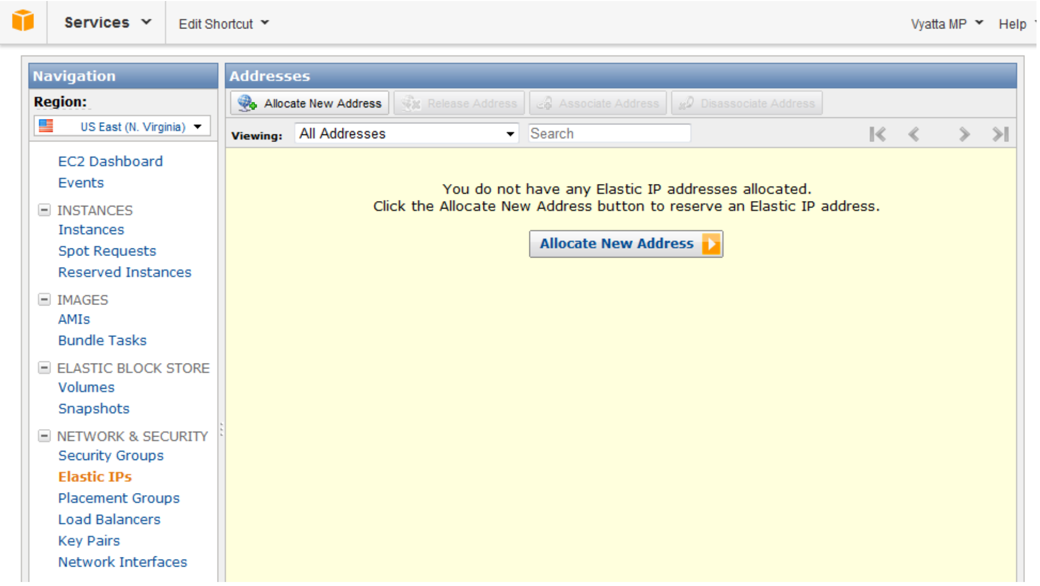
Assigning an AWS Elastic IP Address to the Instance

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

To assign an Elastic IP address

1 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

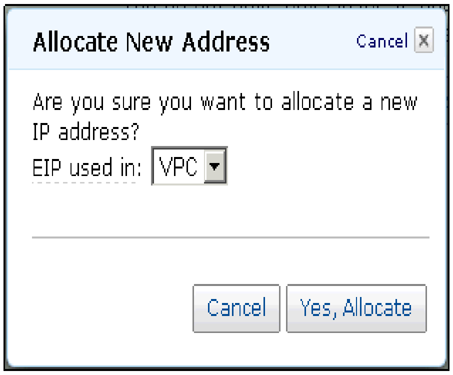
2 In the left navigation pane, select **Elastic IPs**. The **Addresses** pane opens.



Viewing: All Addresses

You do not have any Elastic IP addresses allocated.
Click the Allocate New Address button to reserve an Elastic IP address.

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

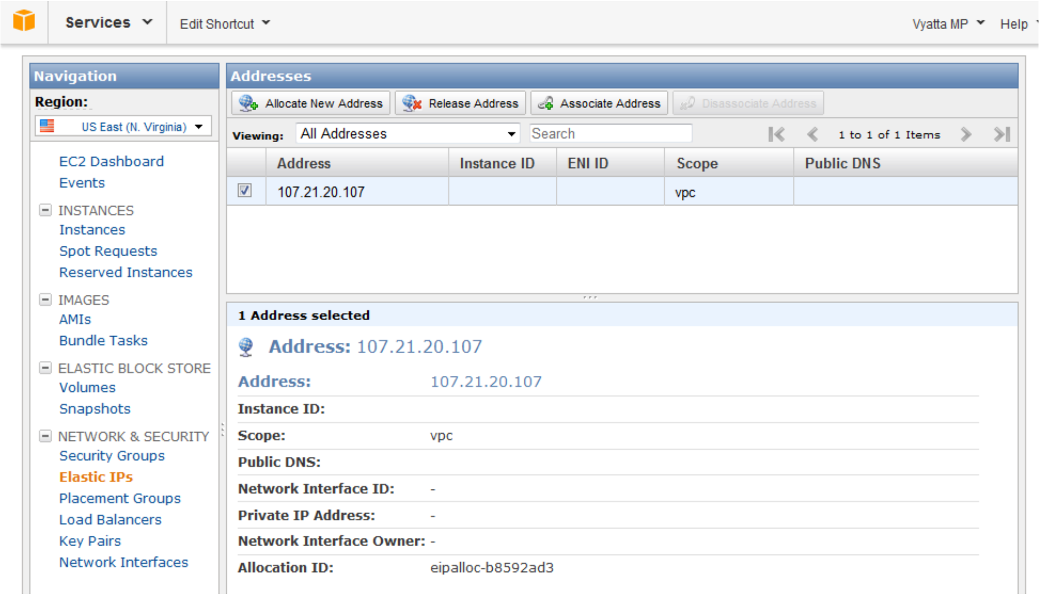


Allocate New Address

Are you sure you want to allocate a new IP address?

EIP used in: VPC

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



Viewing: All Addresses

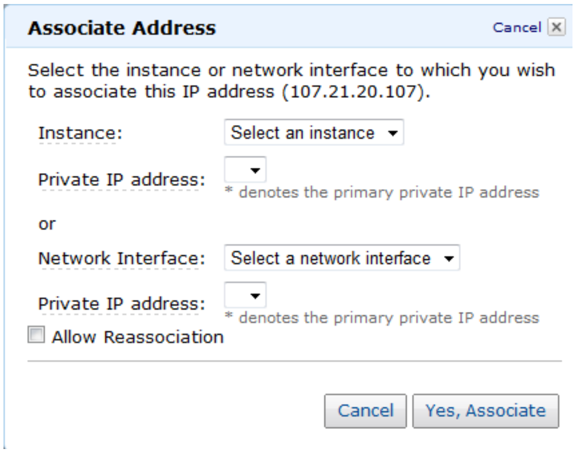
	Address	Instance ID	ENI ID	Scope	Public DNS
☑	107.21.20.107			vpc	

1 Address selected

Address: 107.21.20.107

Address:	107.21.20.107
Instance ID:	
Scope:	vpc
Public DNS:	
Network Interface ID:	-
Private IP Address:	-
Network Interface Owner:	-
Allocation ID:	eipalloc-b8592ad3

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



Select the instance or network interface to which you wish to associate this IP address (107.21.20.107).

Instance: Select an instance

Private IP address:
* denotes the primary private IP address

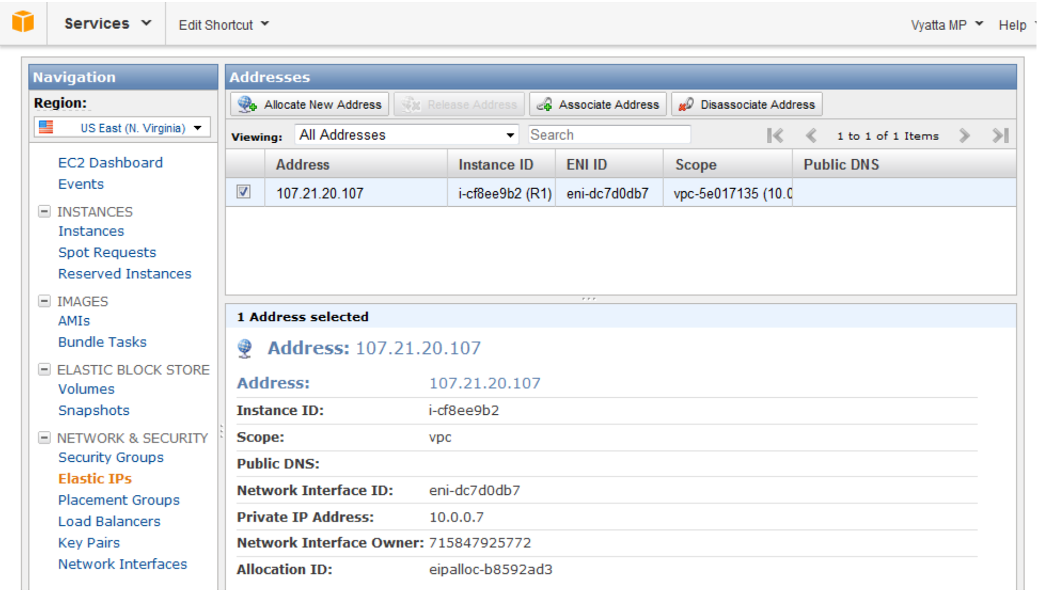
or

Network Interface: Select a network interface

Private IP address:
* denotes the primary private IP address

□ Allow Reassociation

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** pane.



	Address	Instance ID	ENI ID	Scope	Public DNS
☑	107.21.20.107	i-cf8ee9b2 (R1)	eni-dc7d0db7	vpc-5e017135 (10.0	

1 Address selected

Address: 107.21.20.107

Address:	107.21.20.107
Instance ID:	i-cf8ee9b2
Scope:	vpc
Public DNS:	
Network Interface ID:	eni-dc7d0db7
Private IP Address:	10.0.0.7
Network Interface Owner:	715847925772
Allocation ID:	eipalloc-b8592ad3

Accessing 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**. You will be prompted to change the password.

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 to a VPC with a Virtual Tunnel Interface
- 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

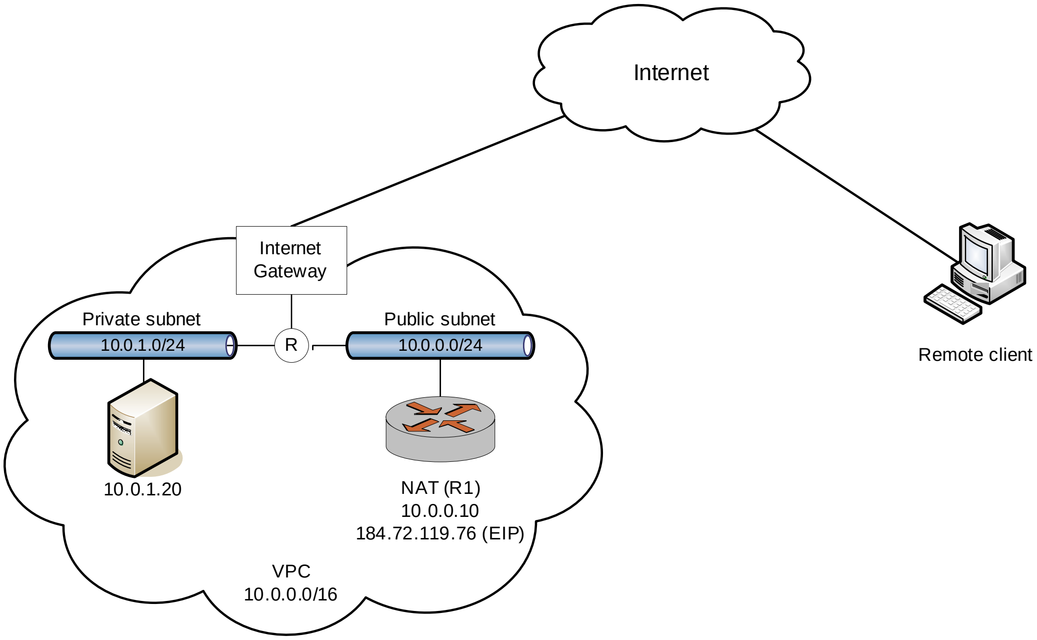
At the end of the installation procedure described in Chapter 1: Installing the System, the following prerequisites for the examples in this chapter were completed:

- 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 and ICMP traffic.
- 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 nat source rule 10 outbound-interface eth0
[edit]
vyatta@vyatta# set nat source rule 10 source address 10.0.1.0/24
[edit]
vyatta@vyatta# set nat source rule 10 translation address 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 nat destination rule 20 destination port 3333
[edit]
vyatta@vyatta# set nat destination rule 20 inbound-interface eth0
[edit]
vyatta@vyatta# set nat destination rule 20 translation address 10.0.1.20
[edit]
vyatta@vyatta# set nat destination rule 20 translation port 22
[edit]
vyatta@vyatta# set nat destination rule 20 protocol tcp
[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 nat
 destination {
     rule 20 {
         destination {
             port 3333
         }
         inbound-interface eth0
         protocol tcp
         translation {
             address 10.0.1.20
             port 22
         }
     }
 }
 source {
     rule 10 {
         outbound-interface eth0
         source {
             address 10.0.1.0/24
         }
         translation {
             address masquerade
         }
     }
 }
[edit]
vyatta@vyatta#
```

8 Exit configuration mode and then exit the login session.

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

```
logout
```

The SSH session terminates.

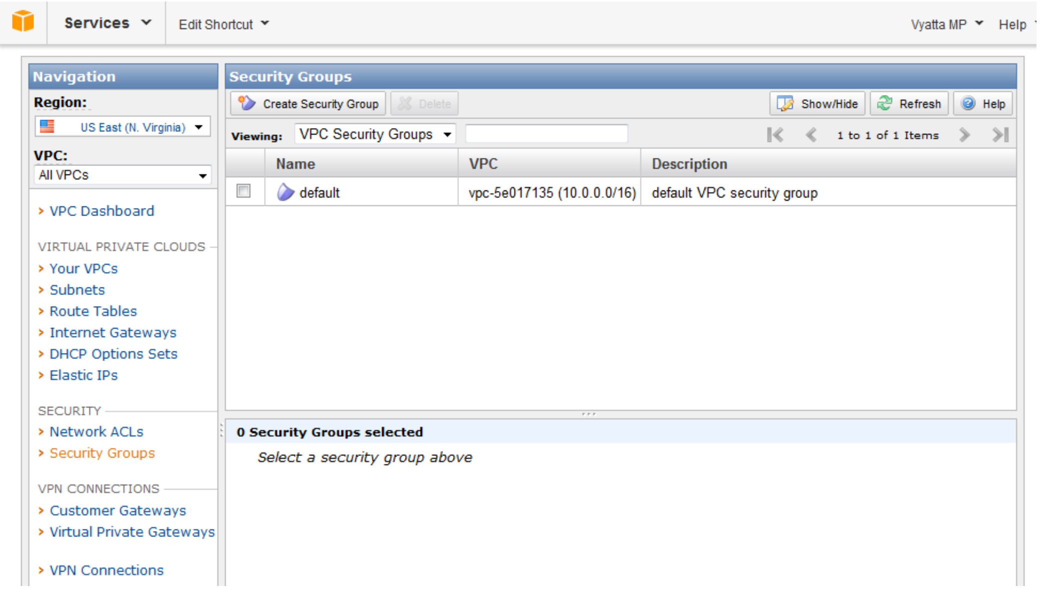
Modify the Default Security Group

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 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

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



Region: US East (N. Virginia)

VPC: All VPCs

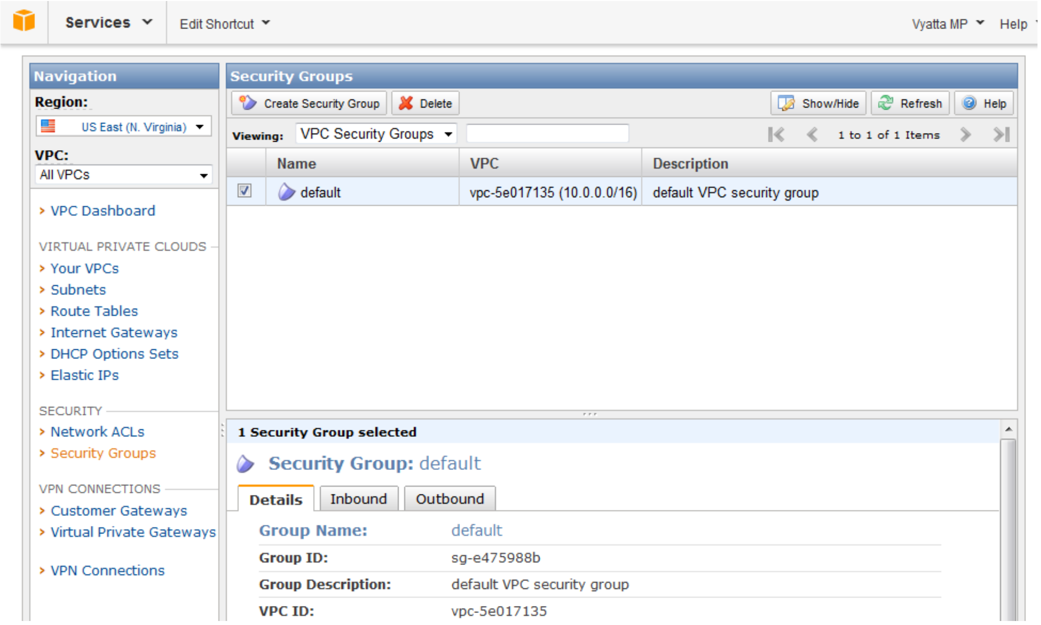
Viewing: VPC Security Groups

	Name	VPC	Description
□	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

0 Security Groups selected

Select a security group above

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



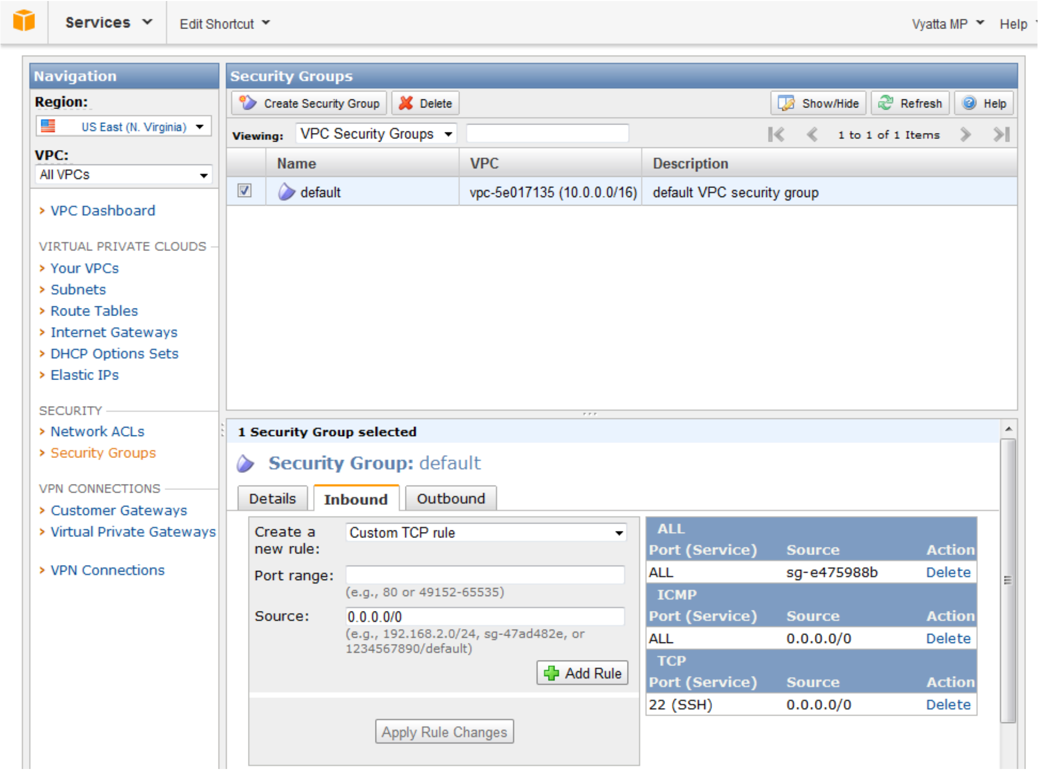
	Name	VPC	Description
☑	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

1 Security Group selected

Security Group: default

Group Name:	default
Group ID:	sg-e475988b
Group Description:	default VPC security group
VPC ID:	vpc-5e017135

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



	Name	VPC	Description
☑	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

1 Security Group selected

Security Group: default

Create a new rule: Custom TCP rule

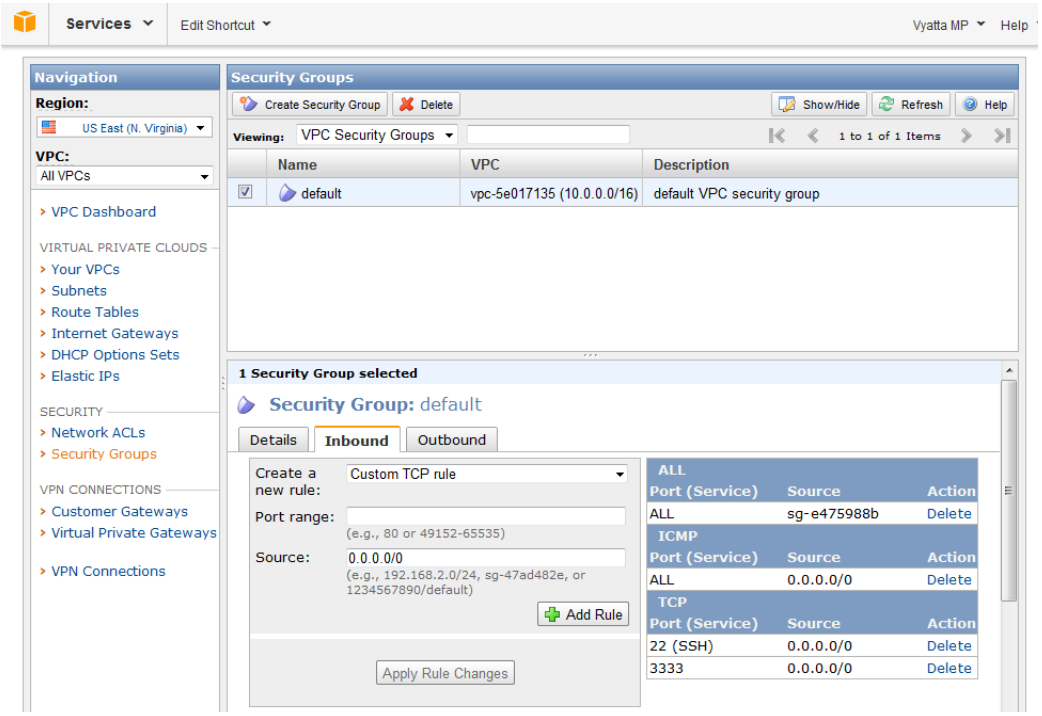
Port range:
(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

Port (Service)	Source	Action
ALL		
ALL	sg-e475988b	Delete
ICMP		
ALL	0.0.0.0/0	Delete
TCP		
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.



☑	Name	VPC	Description
☑	default	vpc-5e017135 (10.0.0.0/16)	default VPC security group

Create a new rule: Custom TCP rule

Port range:
(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

Port (Service)	Source	Action
ALL		
ALL	sg-e475988b	Delete
ICMP		
ALL	0.0.0.0/0	Delete
TCP		
22 (SSH)	0.0.0.0/0	Delete
3333	0.0.0.0/0	Delete

Allow the Instance to Be Used for NAT

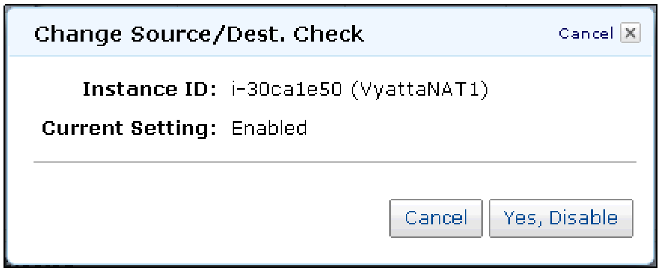
In order for the instance to be used as a NAT device, source and destination address checking must be disabled.

To disable source and destination address checking:

1 Click **EC2** on the AWS Management Console Home page. The **Amazon EC2 Console Dashboard** page appears.

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.



Instance ID: i-30ca1e50 (VyattaNAT1)

Current Setting: Enabled

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

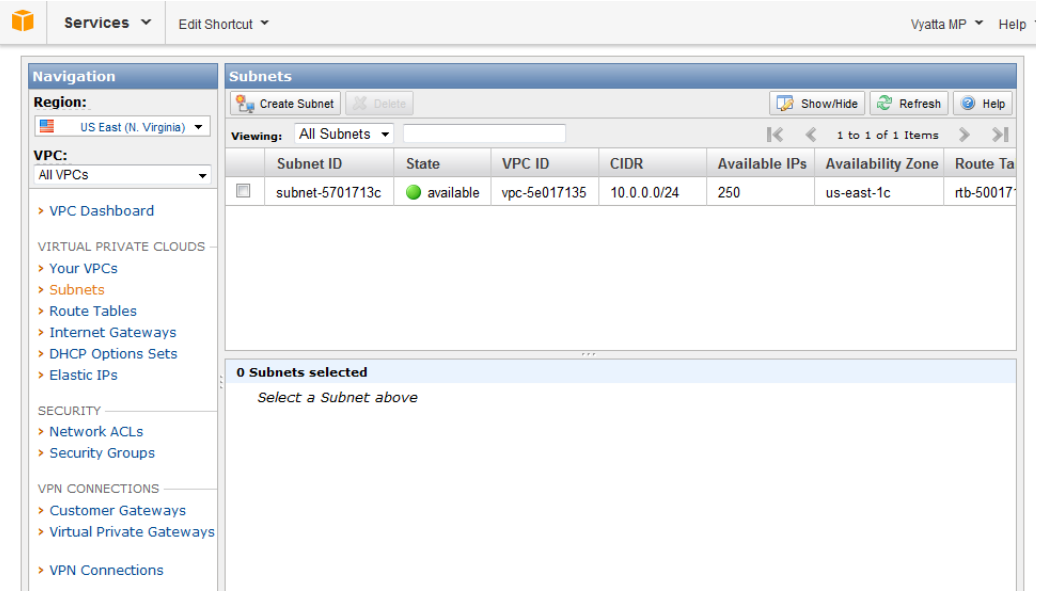
Create a Private Subnet

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

To create a private subnet:

1 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

2 On the left navigation pane, select **Subnets**. The **Subnets** page opens.



Region: US East (N. Virginia)

VPC: All VPCs

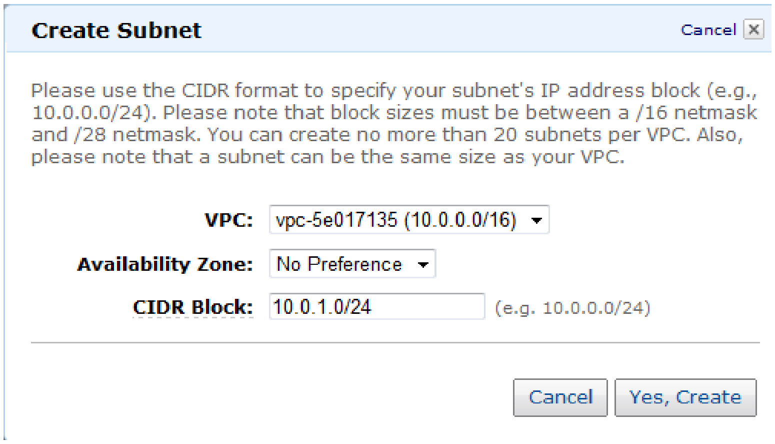
Viewing: All Subnets

	Subnet ID	State	VPC ID	CIDR	Available IPs	Availability Zone	Route Ta
□	subnet-5701713c	● available	vpc-5e017135	10.0.0.0/24	250	us-east-1c	rtb-50017

0 Subnets selected

Select a Subnet above

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



Create Subnet

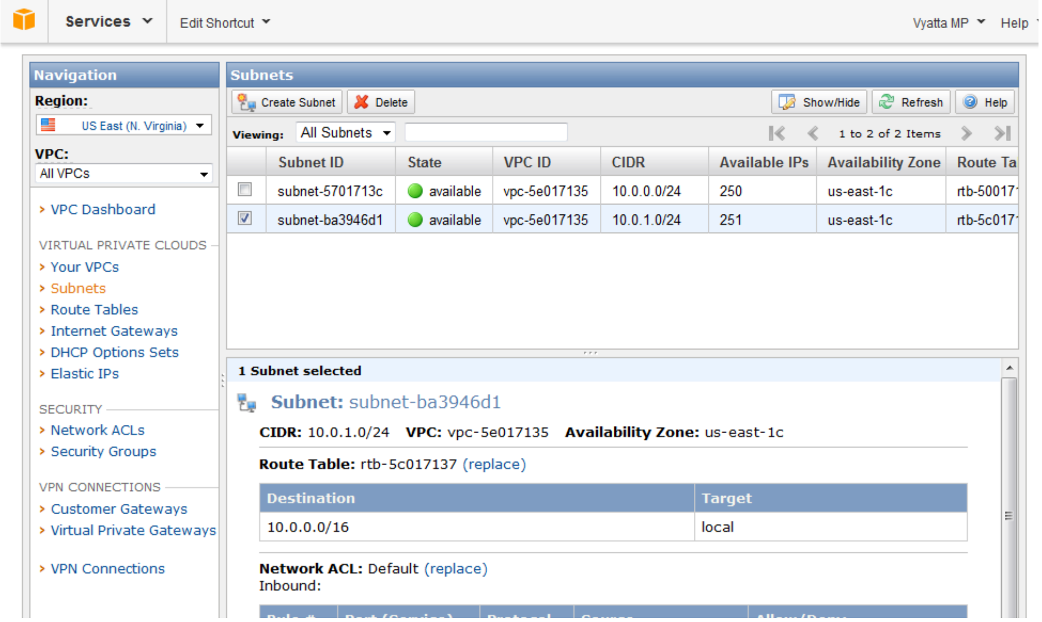
Please use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Please note that block sizes must be between a /16 netmask and /28 netmask. You can create no more than 20 subnets per VPC. Also, please note that a subnet can be the same size as your VPC.

VPC:	vpc-5e017135 (10.0.0.0/16)
Availability Zone:	No Preference
CIDR Block:	10.0.1.0/24 (e.g. 10.0.0.0/24)

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.



	Subnet ID	State	VPC ID	CIDR	Available IPs	Availability Zone	Route Ta
☐	subnet-5701713c	● available	vpc-5e017135	10.0.0.0/24	250	us-east-1c	rtb-50017
☑	subnet-ba3946d1	● available	vpc-5e017135	10.0.1.0/24	251	us-east-1c	rtb-5c017

1 Subnet selected

Subnet: subnet-ba3946d1

CIDR: 10.0.1.0/24 **VPC:** vpc-5e017135 **Availability Zone:** us-east-1c

Route Table: rtb-5c017137 (replace)

Destination	Target
10.0.0.0/16	local

Network ACL: Default (replace)
Inbound:

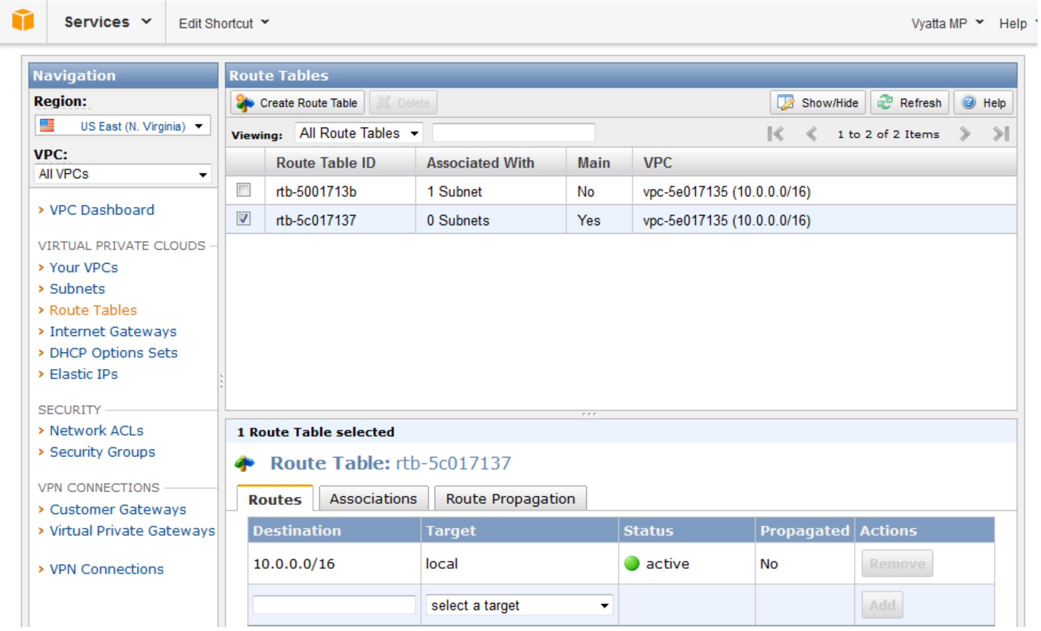
Associate a Route Table with the Private Subnet

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.

To associate a route table with the private subnet:

1 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

2 In the left navigation pane, select **Route Tables**. The **Route Tables** page opens.



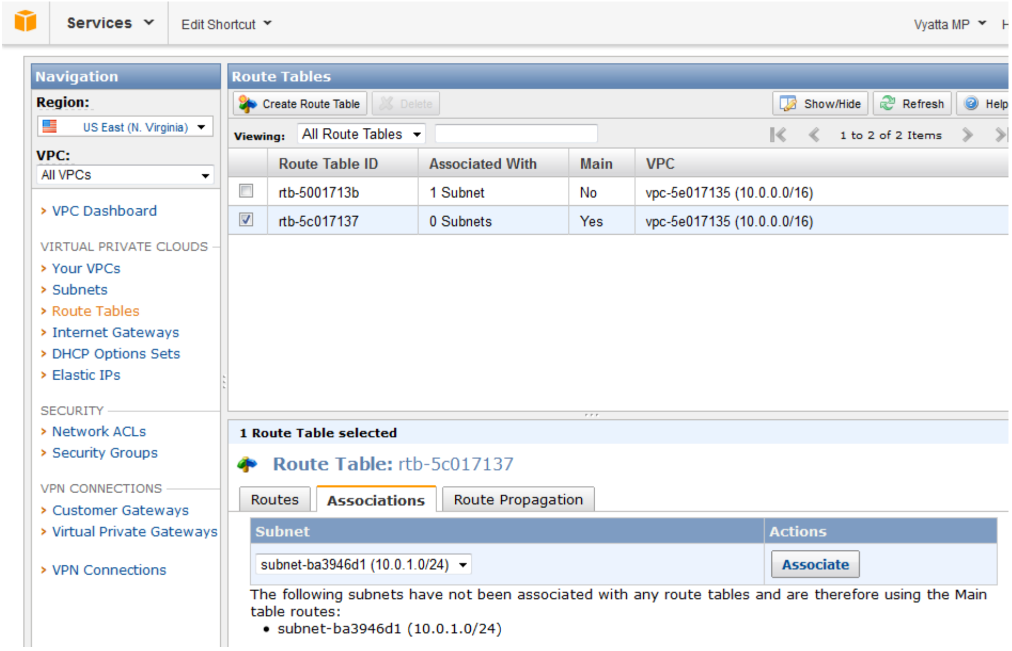
	Route Table ID	Associated With	Main	VPC
□	rtb-5001713b	1 Subnet	No	vpc-5e017135 (10.0.0.0/16)
☑	rtb-5c017137	0 Subnets	Yes	vpc-5e017135 (10.0.0.0/16)

1 Route Table selected

Route Table: rtb-5c017137

Destination	Target	Status	Propagated	Actions
10.0.0.0/16	local	● active	No	Remove
	select a target			Add

3 Select the route table that was created when you created the new subnet and click the **Associations** tab. The **Associations** tab opens.



	Route Table ID	Associated With	Main	VPC
□	rtb-5001713b	1 Subnet	No	vpc-5e017135 (10.0.0.0/16)
☑	rtb-5c017137	0 Subnets	Yes	vpc-5e017135 (10.0.0.0/16)

1 Route Table selected

Route Table: rtb-5c017137

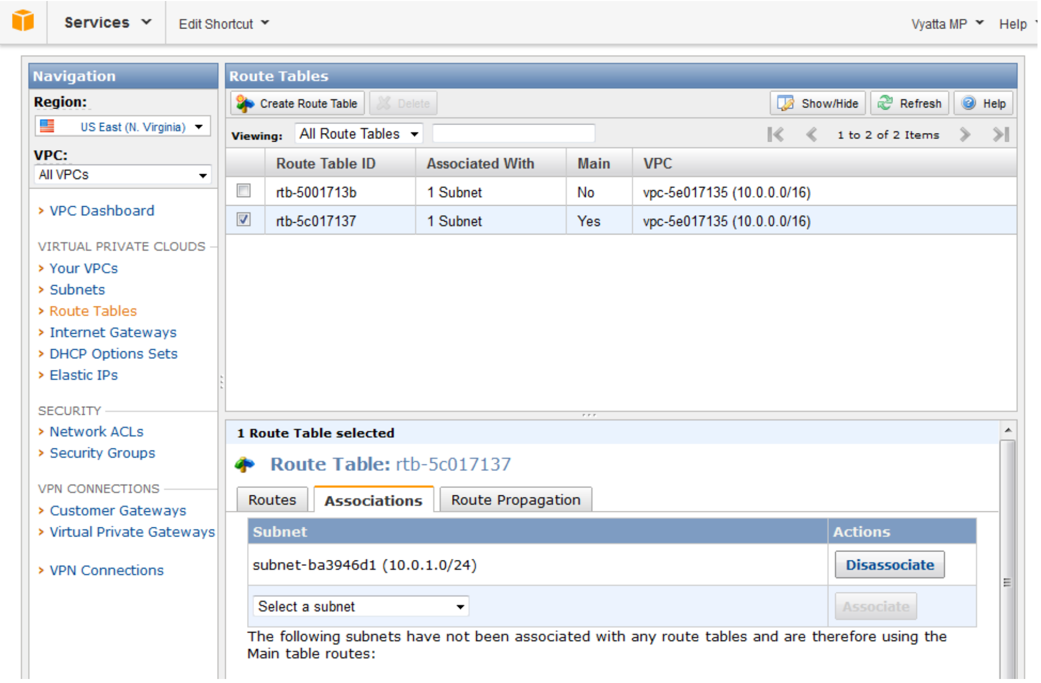
Subnet	Actions
subnet-ba3946d1 (10.0.1.0/24)	Associate

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

- subnet-ba3946d1 (10.0.1.0/24)

4 In the **Subnet** field, select the subnet that was just created (in this case **10.0.1.0/24**) and click the **Associate** button. The Associate Route Table dialog appears.

5 In the **Associate Route Table** dialog, click **Yes, Associate.** The route table is associated with the **10.0.1.0/24** subnet.



	Route Table ID	Associated With	Main	VPC
□	rtb-5001713b	1 Subnet	No	vpc-5e017135 (10.0.0.0/16)
☑	rtb-5c017137	1 Subnet	Yes	vpc-5e017135 (10.0.0.0/16)

1 Route Table selected

Route Table: rtb-5c017137

Subnet	Actions
subnet-ba3946d1 (10.0.1.0/24)	Disassociate
Select a subnet	Associate

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

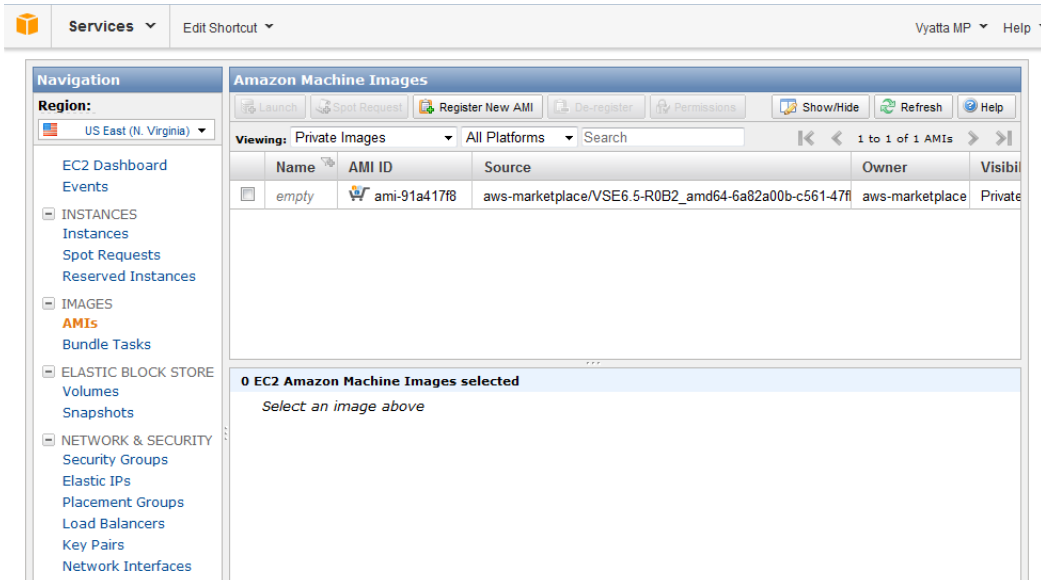
Launch an Instance into the Private Subnet

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. This example assumes that the Vyatta AMI is obtained from the EC2 Console, but it could also be obtained from the AWS Marketplace.

To launch a Vyatta AMI instance into the private subnet

1 Click **EC2** on the AWS Management Console Home page. The **Amazon EC2 Console Dashboard** page appears.

2 In the left navigation pane, select **AMIs**. The **Amazon Machine Images** page opens on the right.



Services Edit Shortcut Vyatta MP Help

Navigation

Region: US East (N. Virginia)

EC2 Dashboard
Events

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
Network Interfaces

Amazon Machine Images

Launch | Spot Request | Register New AMI | De-register | Permissions | Show/Hide | Refresh | Help

Viewing: Private Images | All Platforms | Search — 1 to 1 of 1 AMIs

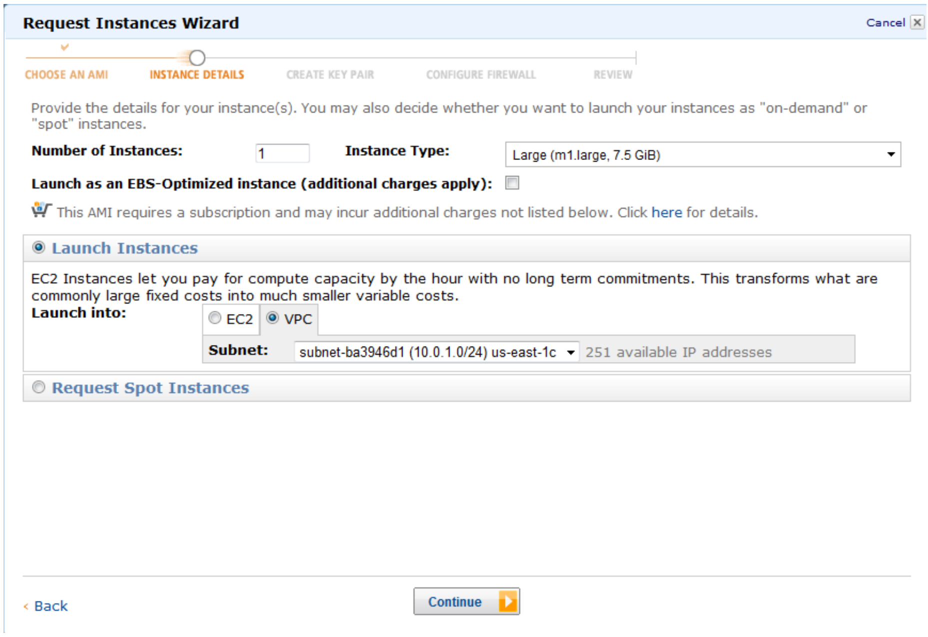
	Name	AMI ID	Source	Owner	Visibil
□	empty	ami-91a417f8	aws-marketplace/VSE6.5-R0B2_amd64-6a82a00b-c561-47f	aws-marketplace	Private

0 EC2 Amazon Machine Images selected

Select an image above

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.



Request Instances Wizard Cancel ☒

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

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** Large (m1.large, 7.5 GiB)

Launch as an EBS-Optimized instance (additional charges apply): □

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

● Launch Instances

EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

Launch into: ○ EC2 ● VPC

Subnet: subnet-ba3946d1 (10.0.1.0/24) us-east-1c 251 available IP addresses

○ Request Spot Instances

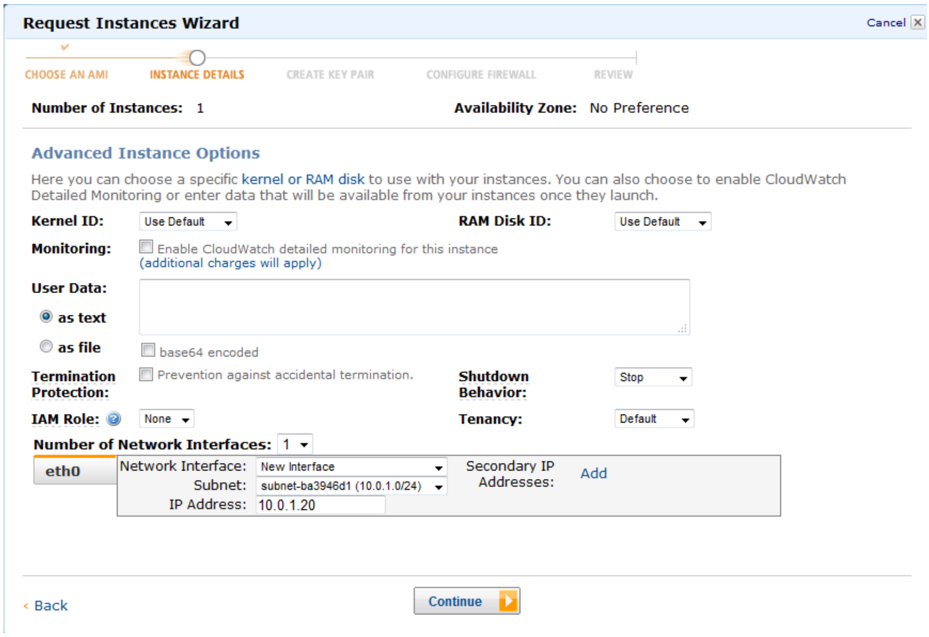
‹ Back Continue

5 Select **Large** (**m1.large**) 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 In the **Launch Instances** section, select **VPC.**

7 In the **Subnet:** 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

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

Termination Protection: □ Prevention against accidental termination.

Shutdown Behavior: Stop

IAM Role: None

Tenancy: Default

Number of Network Interfaces: 1

eth0

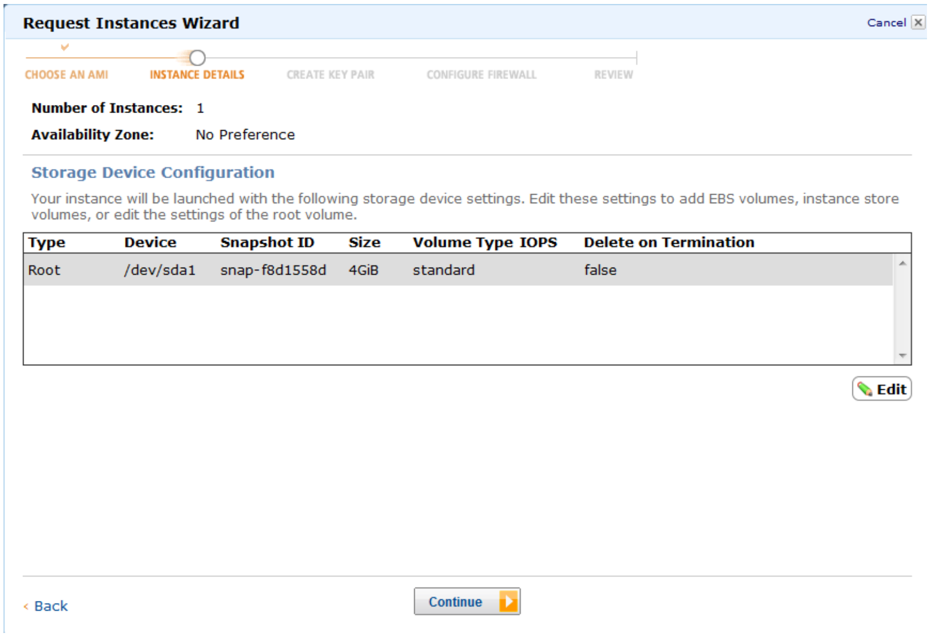
Network Interface: New Interface

Subnet: subnet-ba3946d1 (10.0.1.0/24)

IP Address: 10.0.1.20

Secondary IP Addresses: Add

8 In the **IP Address:** field, enter **10.0.1.20** and click **Continue**. The **Storage Device Configuration** page opens.



Number of Instances: 1

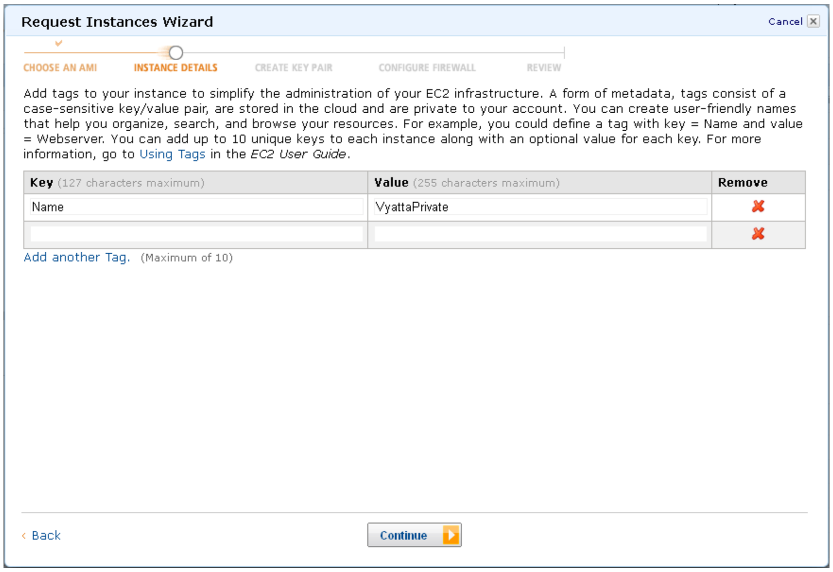
Availability Zone: No Preference

Storage Device Configuration

Your instance will be launched with the following storage device settings. Edit these settings to add EBS volumes, instance store volumes, or edit the settings of the root volume.

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination
Root	/dev/sda1	snap-f8d1558d	4GiB	standard		false

9 The **Add Tags** page appears. If you wish to change the size of the storage deveice associated with the instance, click the **Edit** button. In most cases, this is not necessary. Click **Continue**. The add key page opens.

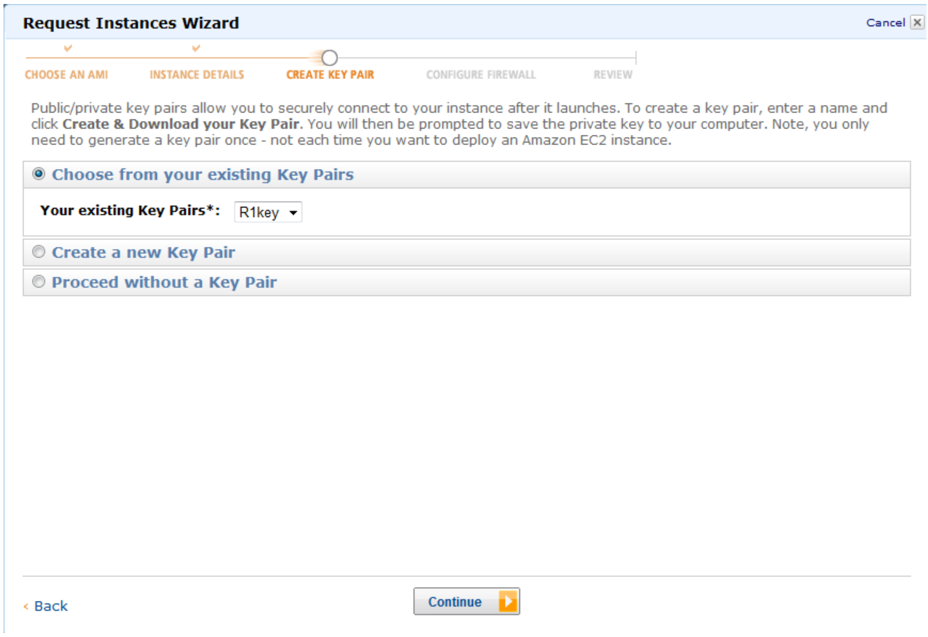


Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. 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	✗
		✗

Add another Tag. (Maximum of 10)

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



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.

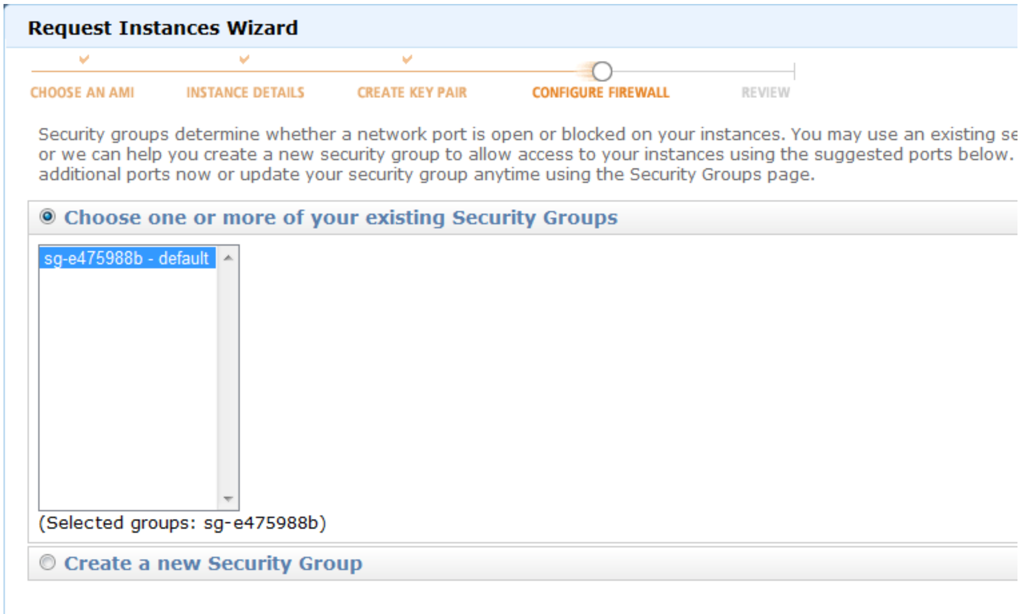
● Choose from your existing Key Pairs

Your existing Key Pairs*: R1key

○ Create a new Key Pair

○ Proceed without a Key Pair

11 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.



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

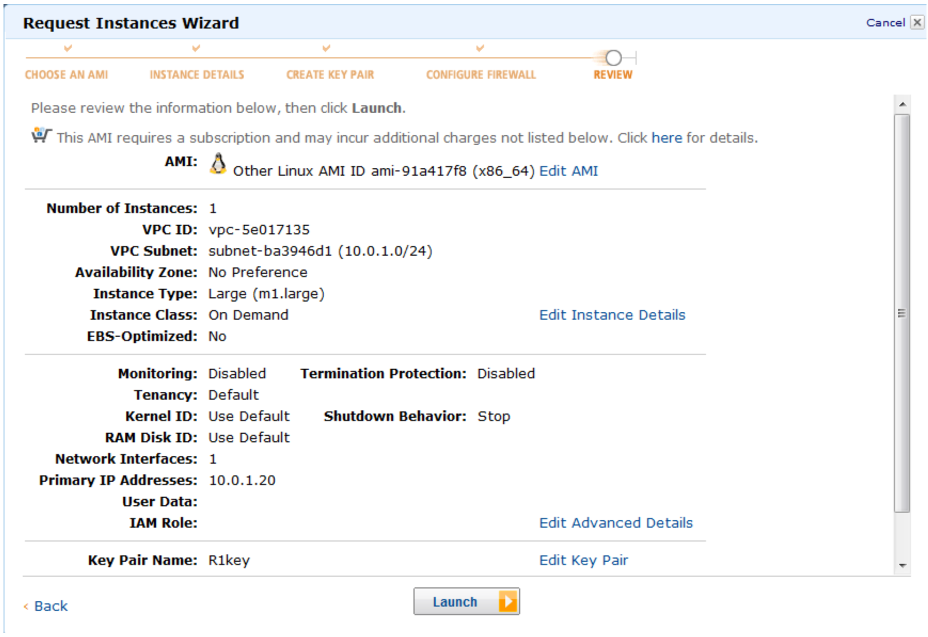
● Choose one or more of your existing Security Groups

sg-e475988b - default

(Selected groups: sg-e475988b)

○ Create a new Security Group

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



Request Instances Wizard

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

This AMI requires a subscription and may incur additional charges not listed below. Click here for details.

AMI: Other Linux AMI ID ami-91a417f8 (x86_64) Edit AMI

Number of Instances: 1
VPC ID: vpc-5e017135
VPC Subnet: subnet-ba3946d1 (10.0.1.0/24)
Availability Zone: No Preference
Instance Type: Large (m1.large)
Instance Class: On Demand
EBS-Optimized: No

Edit Instance Details

Monitoring: Disabled **Termination Protection:** Disabled
Tenancy: Default
Kernel ID: Use Default **Shutdown Behavior:** Stop
RAM Disk ID: Use Default
Network Interfaces: 1
Primary IP Addresses: 10.0.1.20
User Data:
IAM Role:

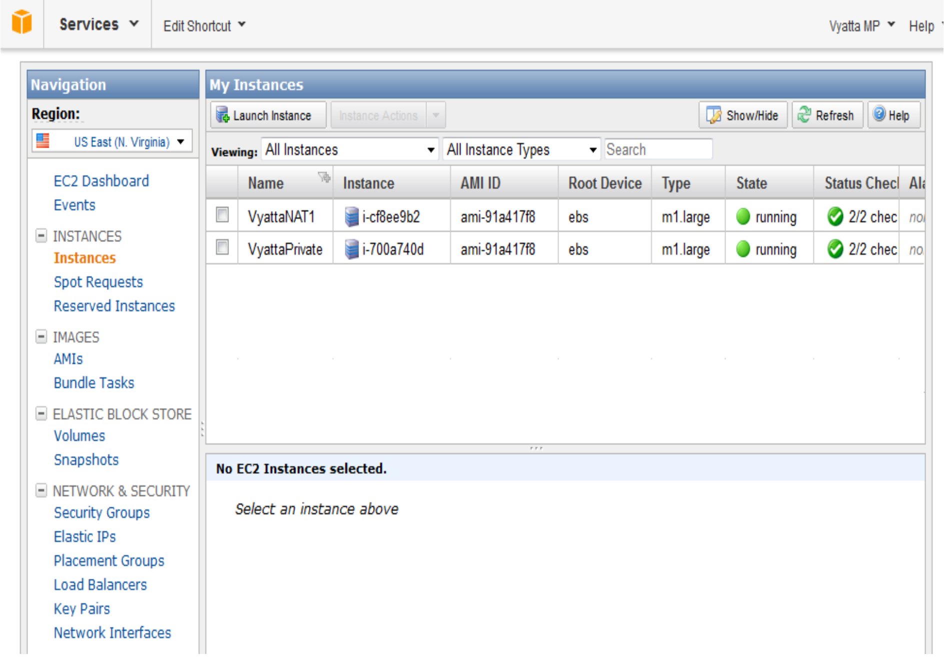
Edit Advanced Details

Key Pair Name: R1key

Edit Key Pair

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

14 To view the status of the newly launched instance, select **Instances** on the left navigation pane within the **EC2** tab.



Viewing: All Instances | All Instance Types | Search

	Name	Instance	AMI ID	Root Device	Type	State	Status Checl	Al
□	VyattaNAT1	i-cf8ee9b2	ami-91a417f8	ebs	m1.large	● running	✓ 2/2 chec	no
□	VyattaPrivate	i-700a740d	ami-91a417f8	ebs	m1.large	● running	✓ 2/2 chec	no

No 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.

To confirm that the instance 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
vyatta@vyatta:~$
```

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).

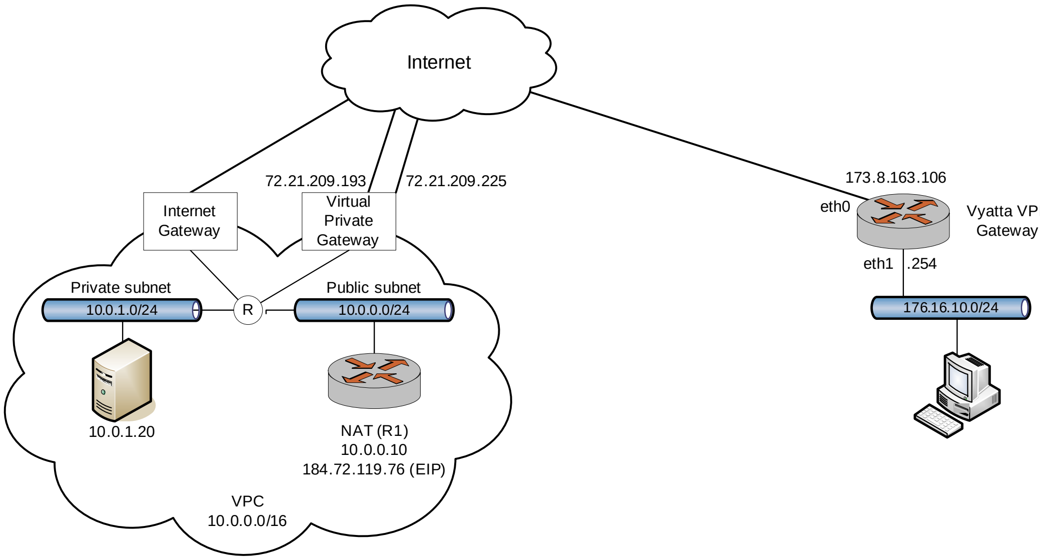
```
vyatta@vyatta:~$ ping www.vyatta.com
PING www.vyatta.com (76.74.103.45) 56(84) bytes of data.
64 bytes from www.vyatta.com (76.74.103.45): icmp_req=1 ttl=46 time=74.4
ms
64 bytes from www.vyatta.com (76.74.103.45): icmp_req=2 ttl=46 time=74.5
ms
^C
--- www.vyatta.com ping statistics ---
2 packets 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 to a VPC with a Virtual Tunnel Interface

This section presents the following topics:

- Configure the VPC
- Configure the Vyatta VPN Gateway
- Verify VPN Connectivity
- Configure Routing on the VPC
- Verify End-to-End Connectivity
- Additional Considerations

In this example, two site-to-site IPsec VPN connections are created between a Vyatta VPN Gateway and an AWS Virtual Private Gateway attached to an AWS VPC. The IPsec VPN connections are from 72.21.209.193 and 72.21.209.225 on the Virtual Private Gateway, to 173.8.163.106 on the Vyatta VPN Gateway. Virtual tunnel interfaces are used on the Vyatta device. The IPsec VPN connections use subnets 169.254.255.72/30 and 169.254.255.76/30 and addresses on these subnet are assigned to these interfaces. The example 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 38. It also assumes that an Internet-connected Vyatta system is available to act as the Vyatta VPN Gateway. The following diagram shows the configuration.



Configure the VPC

This section presents the following topics:

- Modify the Security Group
- Create a Customer Gateway
- Create a Virtual Private Gateway
- Attach the Virtual Private Gateway to the VPC
- Create a VPN Connection
- Download the VPN Configuration

Modify the Security Group

To allow inbound Internet Key Exchange (IKE), Encapsulating Security Payload (ESP), and IPsec NAT-T, add three rules to the default VPC security group in the 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). See "Modify the Default Security Group" on page 41 as a reference.

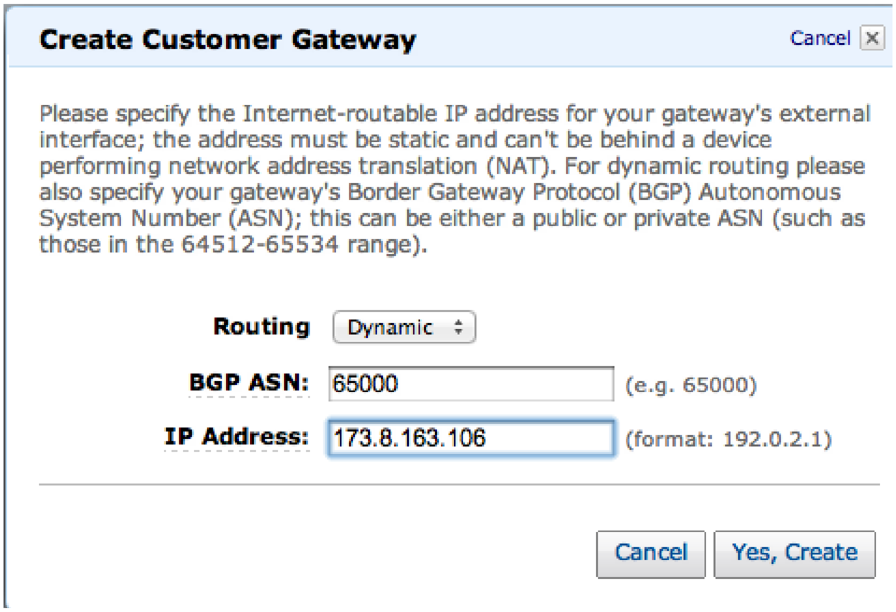
Create a Customer Gateway

The Customer Gateway configuration in the VPC defines how to connect to the Vyatta VPN Gateway.

To create a Customer Gateway

1 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

2 In the left navigation pane, select **Customer Gateways**. The **Create Customer Gateway** dialog opens.



Create Customer Gateway

Please specify the Internet-routable IP address for your gateway's external interface; the address must be static and can't be behind a device performing network address translation (NAT). For dynamic routing please also specify your gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN); this can be either a public or private ASN (such as those in the 64512-65534 range).

Routing: Dynamic

BGP ASN: 65000 (e.g. 65000)

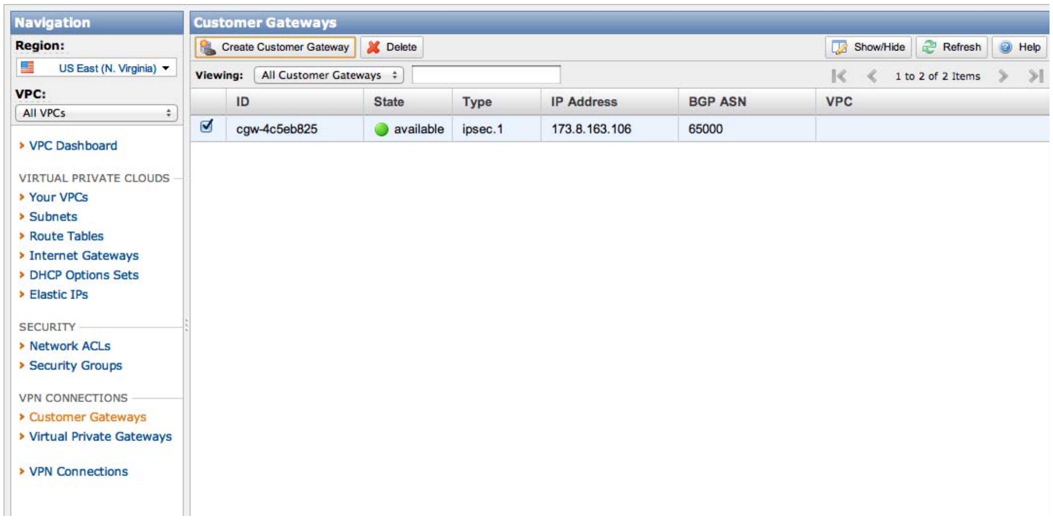
IP Address: 173.8.163.106 (format: 192.0.2.1)

3 In the Routing drop-down menu, select **Dynamic**.

4 In the **BGP ASN** field, enter the BGP Autonomous System Number that the Vyatta VPN Gateway will use. This should be a private ASN if BGP is not in use on this device or within the network. This should be a public ASN if this device is already running BGP with a public ASN or if this device will be advertising the VPC subnet to other BGP speaking routers on the network via iBGP. In this example, we're choosing the default private ASN value of 65000.

5 In the **IP Address** field, enter the public Internet facing IP address of the Vyatta VPN Gateway.

6 Click **Yes, Create.** A newly created Customer Gateway will appear with a State of “available”.



	ID	State	Type	IP Address	BGP ASN	VPC
☑	cgw-4c5eb825	● available	ipsec.1	173.8.163.106	65000	

Create a Virtual Private Gateway

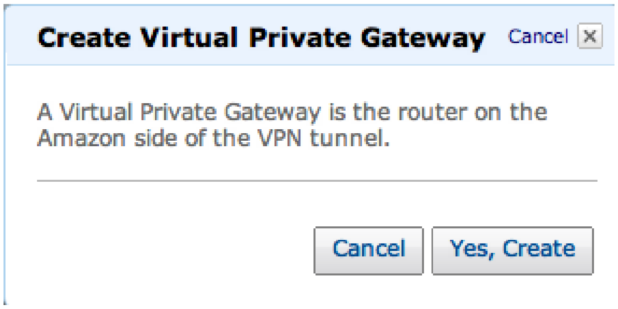
The Virtual Private Gateway is the VPN Gateway on the AWS side of the VPN connection.

To create a Virtual Private Gateway

1 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

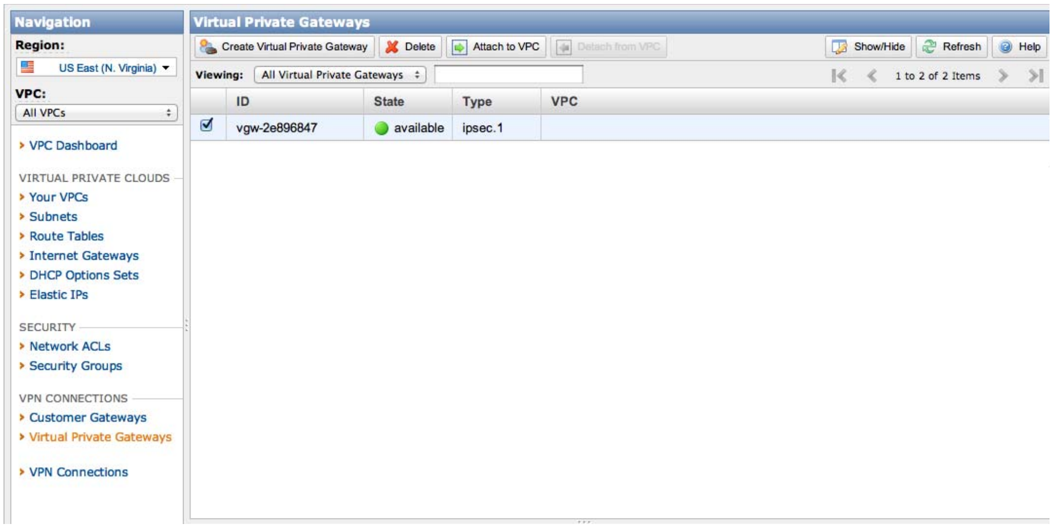
2 In the left navigation pane, select **Virtual Private Gateways.** The **Virtual Private Gateways** page opens.

3 Click **Create Virtual Private Gateway** at the top of the page. The **Create Virtual Private Gateway** dialog opens.



A Virtual Private Gateway is the router on the Amazon side of the VPN tunnel.

4 Click **Yes, Create.** The Virtual Private Gateway is created and configured based on the Customer Gateway configuration. A newly created Virtual Private Gateway will appear with a State of “available”. Take note of the Virtual Private Gateway ID (e.g. vgw-********) for use later when creating the VPN Connection.



Region: US East (N. Virginia)

VPC: All VPCs

Viewing: All Virtual Private Gateways

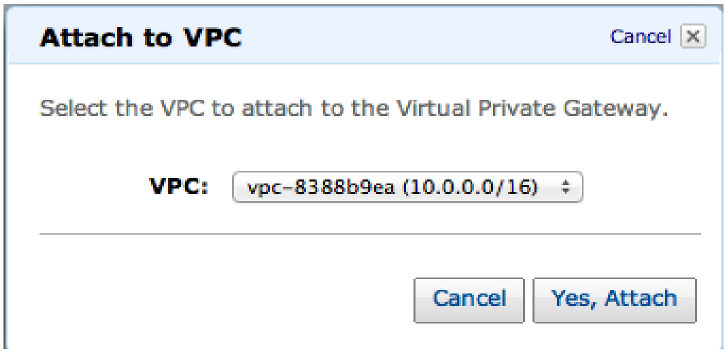
	ID	State	Type	VPC
☑	vgw-2e896847	● available	ipsec.1	

Attach the Virtual Private Gateway to the VPC

The Virtual Private Gateway must be attached to the VPC before it can be used.

To attach the Virtual Private Gateway to the VPC

1 At the top of the Virtual Private Gateways page, click **Attach to VPC**. The **Create Attach to VPC** dialog opens.



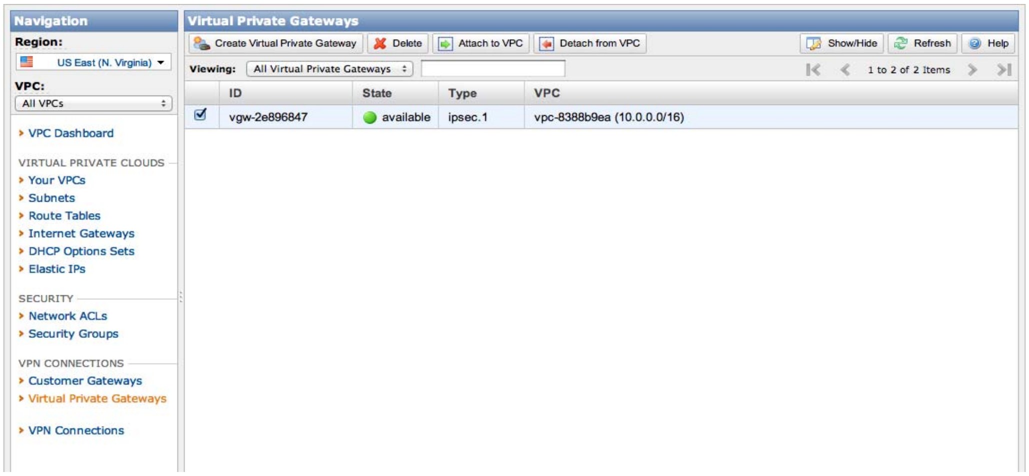
Attach to VPC

Select the VPC to attach to the Virtual Private Gateway.

VPC: vpc-8388b9ea (10.0.0.0/16)

2 In the **VPC** drop-down menu, select the VPC to attach the Virtual Private Gateway to.

3 Click **Yes, Attach.** The VPC appears in the VPC column next to the Virtual Private Gateway attached to it.



Region: US East (N. Virginia)

VPC: All VPCs

Viewing: All Virtual Private Gateways

	ID	State	Type	VPC
☑	vgw-2e896847	● available	ipsec.1	vpc-8388b9ea (10.0.0.0/16)

Create a VPN Connection

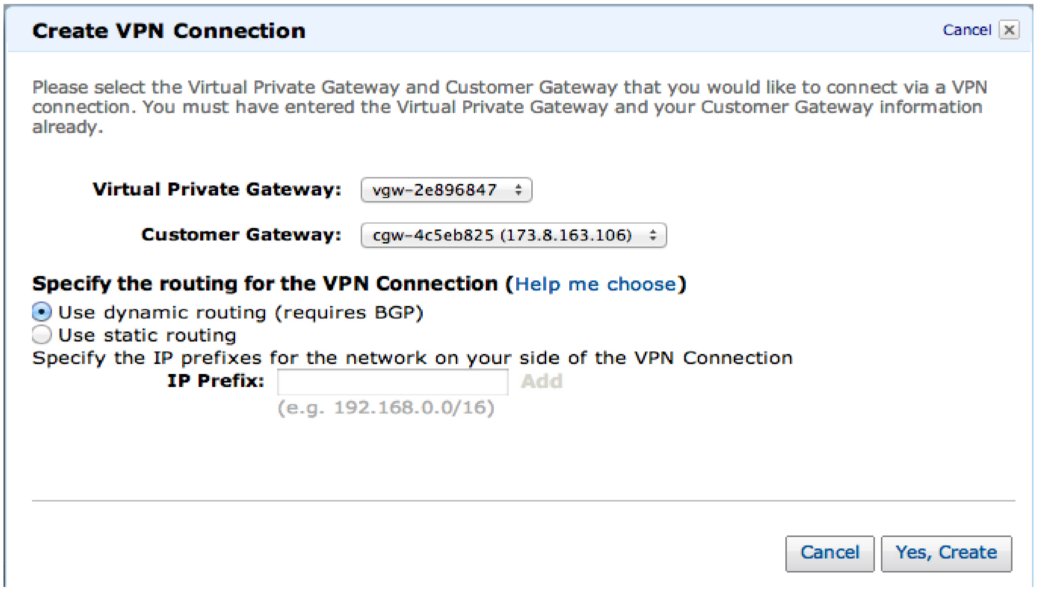
Once the Customer Gateway and Virtual Private Gateway are created and the Virtual Private Gateway is attached to the VPC, a VPN Connection can be created.

To create a VPN Connection

1 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

2 In the left navigation pane, select **VPN Connections**. The **VPN Connections** page opens.

3 Click **Create VPN Connection** at the top of the page. The **Create VPN Connection** dialog opens.



Create VPN Connection

Please select the Virtual Private Gateway and Customer Gateway that you would like to connect via a VPN connection. You must have entered the Virtual Private Gateway and your Customer Gateway information already.

Virtual Private Gateway: vgw-2e896847

Customer Gateway: cgw-4c5eb825 (173.8.163.106)

Specify the routing for the VPN Connection (Help me choose)

● Use dynamic routing (requires BGP)

○ Use static routing

Specify the IP prefixes for the network on your side of the VPN Connection

IP Prefix: (e.g. 192.168.0.0/16)

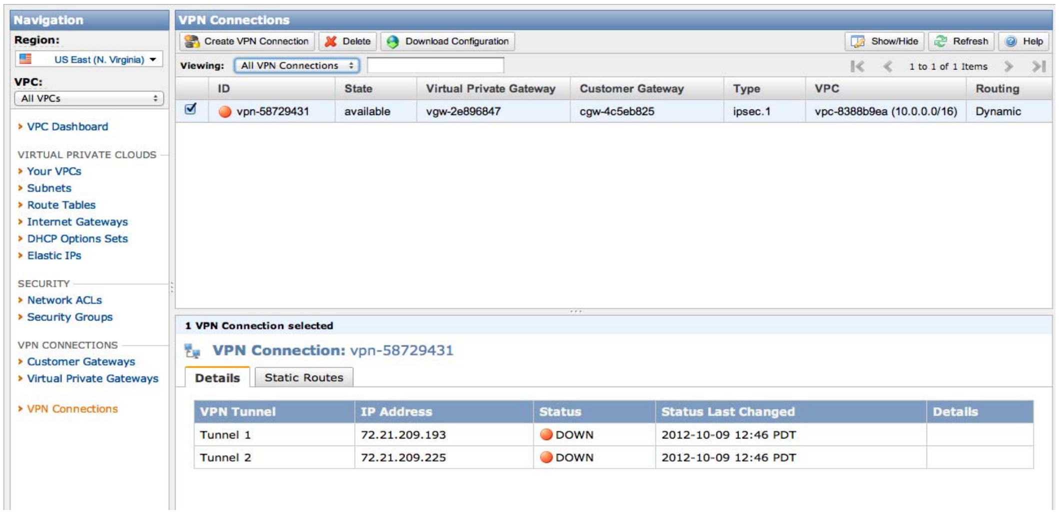
4 In the **Virtual Private Gateway** dropdown menu, select the Virtual Private Gateway created in “Create a Virtual Private Gateway” on page 58.

5 In the **Customer Gateway** dropdown menu, select the Customer Gateway created in “Create a Customer Gateway” on page 56.

6 This example uses dynamic routing. Select **Use dynamic routing (requires BGP)** under the **Specify the routing for the VPN Connection** heading.

7 Click **Yes, Create**. The new VPN Connection appears on the VPN Connections page with a State of “available”.

NOTE *If you select the VPN Connection, you will notice that there are two tunnels created and that both show a Status of DOWN. This is expected. This will be the case until the remote end of the connection (the Vyatta VPN Gateway) is configured and has established a set of VTI tunnels and a BGP peering session with AWS.*



Region: US East (N. Virginia)

VPC: All VPCs

VPN Connections

Viewing: All VPN Connections

1 to 1 of 1 Items

	ID	State	Virtual Private Gateway	Customer Gateway	Type	VPC	Routing
☑	● vpn-58729431	available	vgw-2e896847	cgw-4c5eb825	ipsec.1	vpc-8388b9ea (10.0.0.0/16)	Dynamic

1 VPN Connection selected

VPN Connection: vpn-58729431

VPN Tunnel	IP Address	Status	Status Last Changed	Details
Tunnel 1	72.21.209.193	● DOWN	2012-10-09 12:46 PDT	
Tunnel 2	72.21.209.225	● DOWN	2012-10-09 12:46 PDT	

Download the VPN Configuration

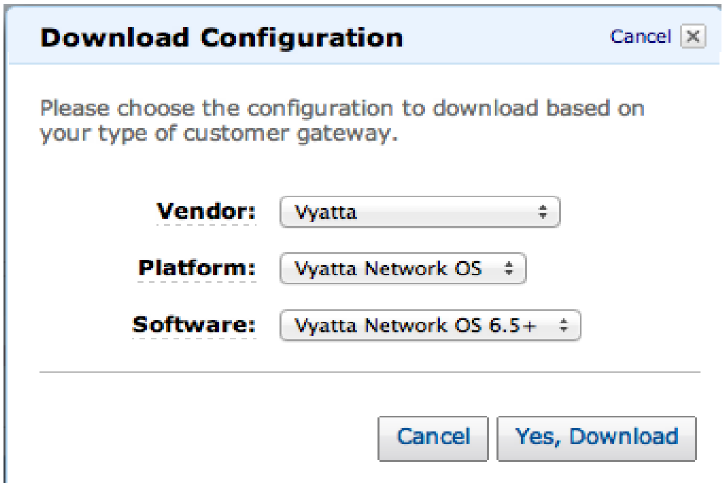
AWS provides a configuration file to assist in configuring the remote end of the VPN for a variety of 3rd part routers, including Vyatta. This file can be retrieved from the VPN Connections page.

To download the VPC Configuration

1 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

2 In the left navigation pane, select **VPN Connections**. The **VPN Connections** page opens.

3 Click **Download Configuration** at the top of the page. The **Download Configuration** dialog opens.



Download Configuration

Please choose the configuration to download based on your type of customer gateway.

Vendor: Vyatta

Platform: Vyatta Network OS

Software: Vyatta Network OS 6.5+

4 In the **Vendor** drop-down menu, select “Vyatta”.

5 In the **Platform** drop-down menu, select “Vyatta Network OS”.

6 In the **Software** drop-down menu, select “Vyatta Network OS 6.5+”.

7 Click **Yes, Download.** Open the downloaded configuration file.

Configure the Vyatta VPN Gateway

The downloaded configuration file contains Vyatta configuration commands that are used to configure the Vyatta VPN Gateway so that it can communicate with the AWS Virtual Private Gateway. These commands must be copied and pasted into a configuration session on the Vyatta VPN Gateway then committed and saved.

To configure the Vyatta VPN Gateway using the downloaded configuration file

1 Login to the Vyatta VPN Gateway.

2 Enter configuration mode by issuing the **configure** command.

3 For each command in the downloaded configuration file, copy the command and paste it into the Vyatta VPN Gateway login session.

NOTE *In order to advertise subnet 172.16.10.0/24 to the VPC, change the command:*
set protocols bgp 65000 network 0.0.0.0/0
to
set protocols bgp 65000 network 172.16.10.0/24

4 When you have copied and pasted all command, enter the **commit** command to commit the configuration, then enter the **save** command to save the configuration. At this point, the pertinent parts of the configuration on the Vyatta VPN Gateway looks as follows:

```
vyatta@R2# show interfaces ethernet
 ethernet eth0 {
     address 173.8.163.106/28
     duplex auto
     hw-id 00:90:fb:1b:85:d9
     smp_affinity auto
     speed auto
 }
 ethernet eth1 {
     address 172.16.10.254/24
     duplex auto
     hw-id 00:90:fb:1b:85:d8
     smp_affinity auto
     speed auto
 }
[edit]
vyatta@R2# show interfaces vti
 vti vti0 {
     address 169.254.255.74/30
     description "VPC tunnel 1"
     mtu 1436
 }
 vti vti1 {
     address 169.254.255.78/30
     description "VPC tunnel 2"
     mtu 1436
 }
[edit]
vyatta@R2# show vpn
 ipsec {
     esp-group AWS {
         compression disable
         lifetime 3600
         mode tunnel
         pfs enable
         proposal 1 {
             encryption aes128
```

```
            hash sha1
        }
    }
    ike-group AWS {
        dead-peer-detection {
            action restart
            interval 15
            timeout 30
        }
        lifetime 28800
        proposal 1 {
            dh-group 2
            encryption aes128
            hash sha1
        }
    }
    ipsec-interfaces {
        interface eth0
    }
    site-to-site {
        peer 72.21.209.193 {
            authentication {
                mode pre-shared-secret
                pre-shared-secret vDoG59257kDPSHjQNFb_ZpPPFFpFx2Nw
            }
            description "VPC tunnel 1"
            ike-group AWS
            local-address 173.8.163.106
            vti {
                bind vti0
                esp-group AWS
            }
        }
        peer 72.21.209.225 {
            authentication {
                mode pre-shared-secret
                pre-shared-secret X9ypqOk2Pcbs1IRWSG9gDb2UYYMUa3pK
            }
            description "VPC tunnel 1"
            ike-group AWS
            local-address 173.8.163.106
            vti {
                bind vti1
                esp-group AWS
            }
        }
    }
}
```

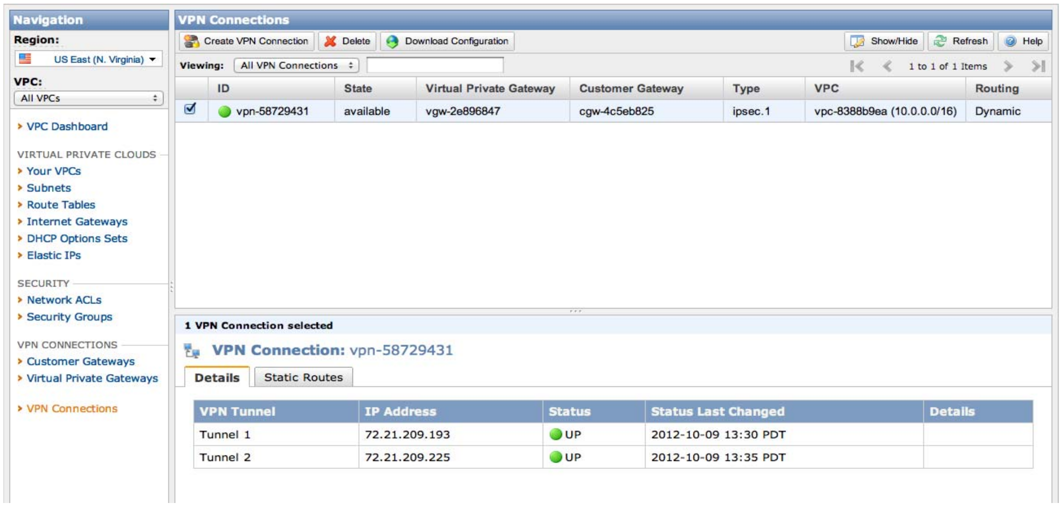
```
[edit]
vyatta@R2# show protocols bgp
 bgp 65000 {
     neighbor 169.254.255.73 {
         remote-as 7224
         soft-reconfiguration {
             inbound
         }
         timers {
             holdtime 30
             keepalive 30
         }
     }
     neighbor 169.254.255.77 {
         remote-as 7224
         soft-reconfiguration {
             inbound
         }
         timers {
             holdtime 30
             keepalive 30
         }
     }
     network 172.16.10.0/24{
     }
 }
[edit]
vyatta@R2#
```

Verify VPN Connectivity

The two VPN tunnels between the AWS Virtual Private Gateway and the Vyatta VPN Gateway should come up at this point. You can verify that the tunnels are up in operational mode on the Vyatta system. The commands and their output should look similar to the following:

```
vyatta@R2:~$ show vpn ipsec sa statistics

Peer ID / IP                            Local ID / IP
------------                            -------------
72.21.209.193                           173.8.163.106

  Description: VPC tunnel 1
  Tunnel Dir Source Network                          Destination Network        Bytes
  ------ --- --------------                          -------------------        -----
  vti    in  0.0.0.0/0                               0.0.0.0/0                  35604
  vti    out 0.0.0.0/0                               0.0.0.0/0                  35703


Peer ID / IP                            Local ID / IP
------------                            -------------
72.21.209.225                           173.8.163.106

  Description: VPC tunnel 1

  Tunnel Dir Source Network                          Destination Network        Bytes
  ------ --- --------------                          -------------------        -----
  vti    in  0.0.0.0/0                                0.0.0.0/0                  6888
  vti    out 0.0.0.0/0                                0.0.0.0/0                  6888


vyatta@R2:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             173.8.163.106/28                  u/u
eth1             172.16.10.254/24                  u/u
lo               127.0.0.1/8                       u/u
                 ::1/128
vti0             169.254.255.74/30                 u/u  VPC tunnel 1
vti1             169.254.255.78/30                 u/u  VPC tunnel 2


vyatta@R2:~$ show ip bgp summary
BGP router identifier 172.16.10.254, local AS number 65000
IPv4 Unicast - max multipaths: ebgp 1 ibgp 1
RIB entries 3, using 192 bytes of memory
Peers 2, using 5048 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
169.254.255.73  4  7224    7712    7655       0    0    0 21:16:06       1
169.254.255.77  4  7224    7685    7627       0    0    0 21:11:20       1

Total number of neighbors 2


vyatta@R2:~$ show ip bgp
BGP table version is 0, local router ID is 172.16.10.254
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  10.0.0.0/16      169.254.255.77                           0 7224 i
*>                  169.254.255.73                           0 7224 i
*> 172.16.10.0/24   0.0.0.0                  1           32768 i

Total number of prefixes 2
```

On the AWS side, on the **VPN Connections** page, you will see that the two tunnels now have a Status of “UP”.



Region: US East (N. Virginia)

VPC: All VPCs

Viewing: All VPN Connections

	ID	State	Virtual Private Gateway	Customer Gateway	Type	VPC	Routing
☑	● vpn-58729431	available	vgw-2e896847	cgw-4c5eb825	ipsec.1	vpc-8388b9ea (10.0.0.0/16)	Dynamic

1 VPN Connection selected

VPN Connection: vpn-58729431

VPN Tunnel	IP Address	Status	Status Last Changed	Details
Tunnel 1	72.21.209.193	● UP	2012-10-09 13:30 PDT	
Tunnel 2	72.21.209.225	● UP	2012-10-09 13:35 PDT	

Configure Routing on the VPC

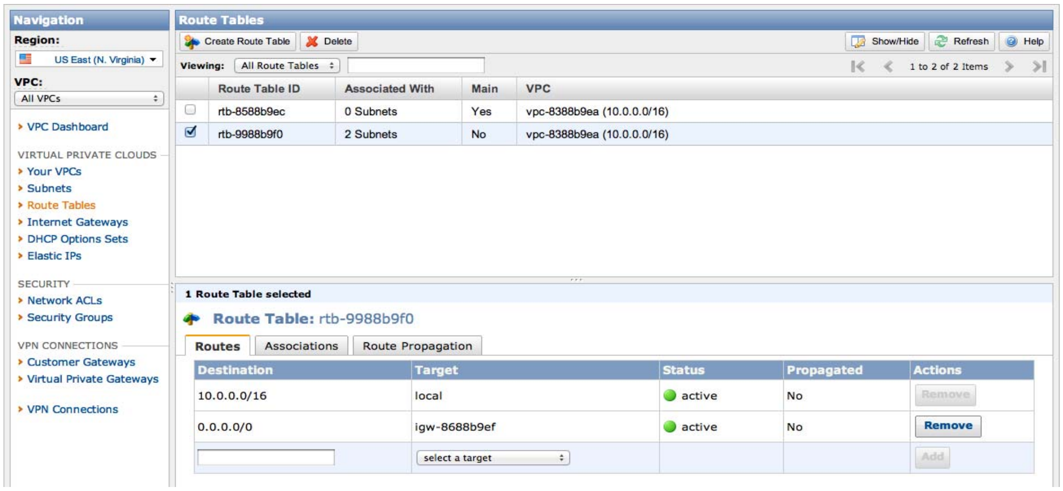
At this point the VPN connections are up but routes learned from BGP are not propogated.

To propogate routes on the VPC

1 Click **VPC** on the AWS Management Console Home page. The **Amazon VPC Console Dashboard** page appears.

2 In the left navigation pane, select **Route Tables**. The **Route Tables** page opens.

3 Select the Route Table that is associated with the VPC subnets that will be reached over the VPN tunnels. More than just one Route Table may need to be updated. The default VPC Route Table is the Main table. Any subnet that is not associated to a Route Table will use the Main table.



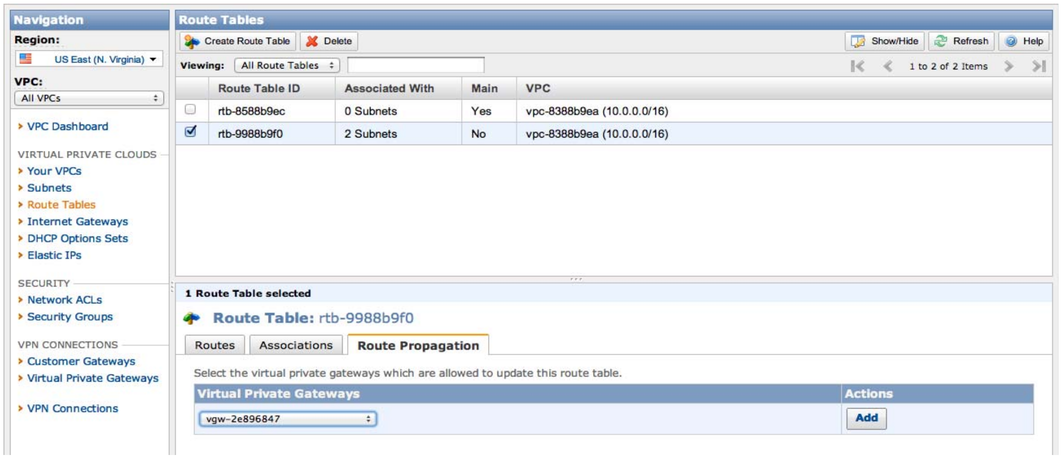
	Route Table ID	Associated With	Main	VPC
□	rtb-8588b9ec	0 Subnets	Yes	vpc-8388b9ea (10.0.0.0/16)
☑	rtb-9988b9f0	2 Subnets	No	vpc-8388b9ea (10.0.0.0/16)

1 Route Table selected

Route Table: rtb-9988b9f0

Destination	Target	Status	Propagated	Actions
10.0.0.0/16	local	● active	No	Remove
0.0.0.0/0	igw-8688b9ef	● active	No	Remove
	select a target			Add

4 Select the **Route Propagation** tab for the selected Route Table.



	Route Table ID	Associated With	Main	VPC
□	rtb-8588b9ec	0 Subnets	Yes	vpc-8388b9ea (10.0.0.0/16)
☑	rtb-9988b9f0	2 Subnets	No	vpc-8388b9ea (10.0.0.0/16)

1 Route Table selected

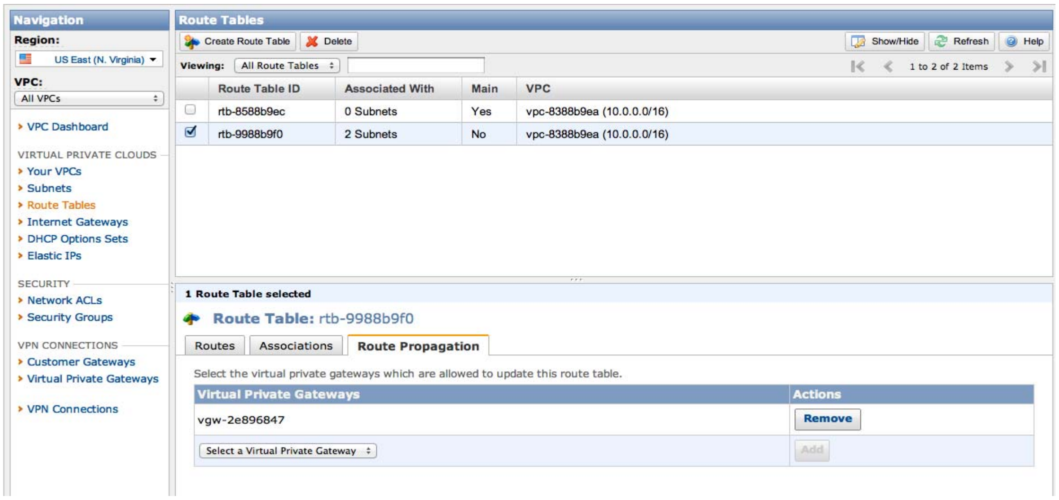
Route Table: rtb-9988b9f0

Select the virtual private gateways which are allowed to update this route table.

Virtual Private Gateways	Actions
vgw-2e896847	Add

5 In the **Virtual Private Gateways** drop-down menu, select the VPN Gateway that was created "Create a Virtual Private Gateway" on page 58.

6 Click **Add**. The Virtual Private Gateway will be added to the list of Virtual Private Gateways that are allowed to update the route table.



	Route Table ID	Associated With	Main	VPC
☐	rtb-8588b9ec	0 Subnets	Yes	vpc-8388b9ea (10.0.0.0/16)
☑	rtb-9988b9f0	2 Subnets	No	vpc-8388b9ea (10.0.0.0/16)

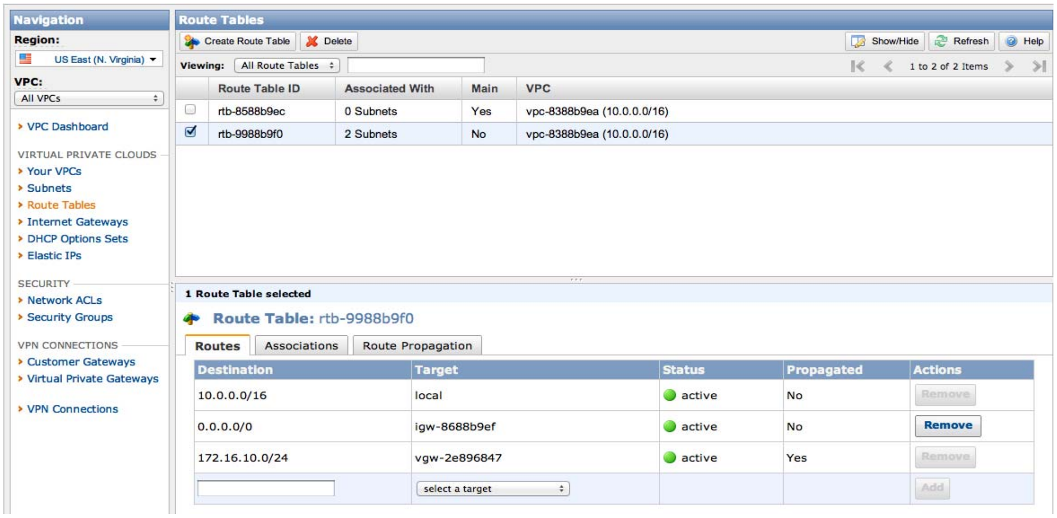
1 Route Table selected

Route Table: rtb-9988b9f0

Select the virtual private gateways which are allowed to update this route table.

Virtual Private Gateways	Actions
vgw-2e896847	Remove

7 Select the Routes tab to view the routes learned from the Virtual Private Gateway.



	Route Table ID	Associated With	Main	VPC
☐	rtb-8588b9ec	0 Subnets	Yes	vpc-8388b9ea (10.0.0.0/16)
☑	rtb-9988b9f0	2 Subnets	No	vpc-8388b9ea (10.0.0.0/16)

1 Route Table selected

Route Table: rtb-9988b9f0

Destination	Target	Status	Propagated	Actions
10.0.0.0/16	local	● active	No	Remove
0.0.0.0/0	igw-8688b9ef	● active	No	Remove
172.16.10.0/24	vgw-2e896847	● active	Yes	Remove

Verify End-to-End Connectivity

Test the configuration by pinging a device in one private subnet (10.0.1.20) from a device in the other private subnet. In this case, we will ping from the Vyatta VPN Gateway but specify that the source address is the IP address associated with eth1 (172.16.10.254).

```
vyatta@R2:~$ ping 10.0.1.20 interface 172.16.10.254
PING 10.0.1.20 (10.0.1.20) from 172.16.10.254 : 56(84) bytes of data.
64 bytes from 10.0.1.20: icmp_req=1 ttl=62 time=99.9 ms
64 bytes from 10.0.1.20: icmp_req=2 ttl=62 time=79.9 ms
64 bytes from 10.0.1.20: icmp_req=3 ttl=62 time=79.9 ms
64 bytes from 10.0.1.20: icmp_req=4 ttl=62 time=89.9 ms
64 bytes from 10.0.1.20: icmp_req=5 ttl=62 time=89.9 ms
^C
--- 10.0.1.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4039ms
rtt min/avg/max/mdev = 79.989/87.987/99.986/7.491 ms
vyatta@R2:~$
```

Additional Considerations

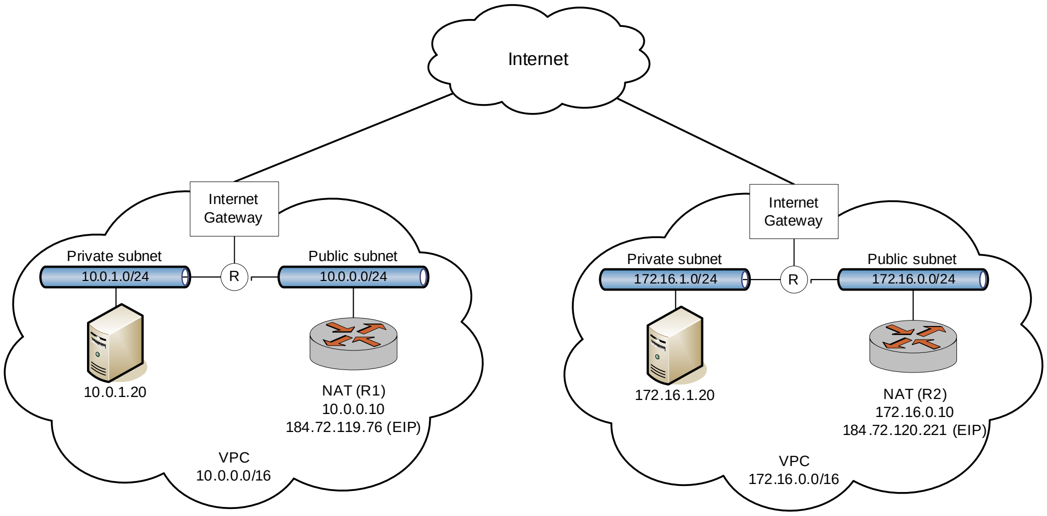
In order to prevent advertising the eBGP learned VPC subnet back to the VPC, you can configure a route-map and apply it to both sessions. The configuration looks as follows:

```
vyatta@R2# show policy
 prefix-list internal {
     rule 10 {
         action permit
         prefix 172.16.10.0/24
     }
 }
 route-map AWS {
     rule 10 {
         action permit
         match {
             ip {
                 address {
                     prefix-list internal
                 }
             }
         }
     }
 }
[edit]
```

```
vyatta@R2# show protocols bgp
 bgp 65000 {
     neighbor 169.254.255.73 {
         remote-as 7224
         route-map {
             export AWS
         }
         soft-reconfiguration {
             inbound
         }
         timers {
             holdtime 30
             keepalive 30
         }
     }
     neighbor 169.254.255.77 {
         remote-as 7224
         route-map {
             export AWS
         }
         soft-reconfiguration {
             inbound
         }
         timers {
             holdtime 30
             keepalive 30
         }
     }
     network 172.16.10.0/24 {
     }
 }
[edit]
```

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 38. The following diagram shows the configuration.



To allow inbound Internet Key Exchange (IKE), Encapsulating Security Payload (ESP), and IPsec NAT-T, add three 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). See "Modify the Default Security Group" on page 41 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 @Router1
                 mode pre-shared-secret
                 pre-shared-secret test_key_1
                 remote-id @Router2
             }
             connection-type initiate
             default-esp-group ESP-1W
             ike-group IKE-1W
             local-address 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 @Router2
                 mode pre-shared-secret
                 pre-shared-secret test_key_1
                 remote-id @Router1
             }
             connection-type initiate
             default-esp-group ESP-1E
             ike-group IKE-1E
             local-address 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).

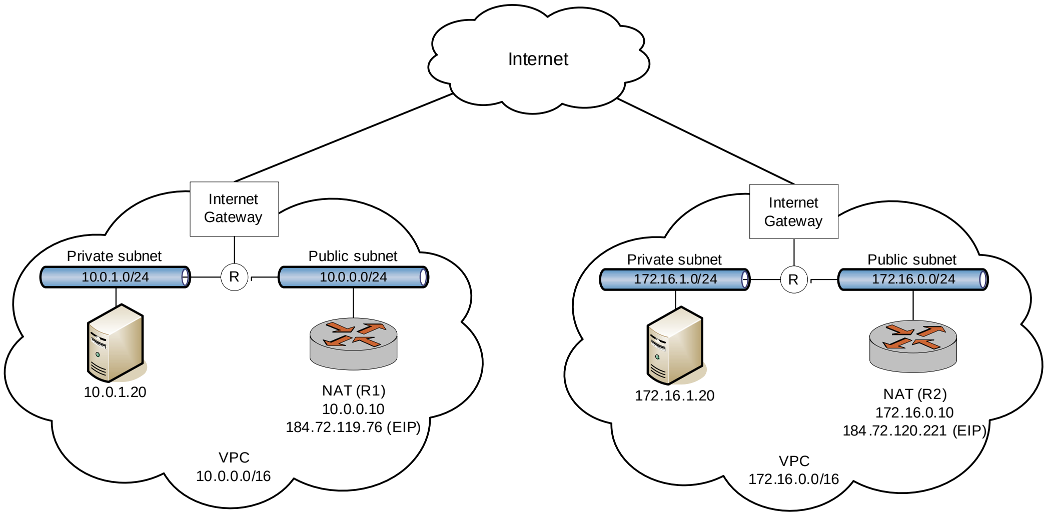
```
vyatta@vyatta:~$ ping 10.0.1.20
PING 10.0.1.20 (10.0.1.20) 56(84) bytes of data.
64 bytes from 10.0.1.20: icmp_req=1 ttl=64 time=0.439 ms
64 bytes from 10.0.1.20: icmp_req=2 ttl=64 time=0.572 ms
64 bytes from 10.0.1.20: icmp_req=3 ttl=64 time=0.430 ms
64 bytes from 10.0.1.20: icmp_req=4 ttl=64 time=0.448 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 38. The following diagram shows the configuration.



To allow inbound OpenVPN traffic, add one rule to the default VPC security group in each VPC. This inbound rule allows UDP traffic on port 1194 from any source (0.0.0.0/0). See "Modify the Default Security Group" on page 41 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 a route on R1 to the remote network via the OpenVPN tunnel, configure it as follows:

```
vyatta@R1# show protocols static
 interface-route 172.16.0.0/16 {
     next-hop-interface vtun0 {
     }
}
[edit]
vyatta@R1#
```

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#
```

To provide a route on R2 to the remote network via the OpenVPN tunnel, configure it as follows:

```
vyatta@R2# show protocols static
 interface-route 10.0.0.0/16 {
     next-hop-interface vtun0 {
     }
}
[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).

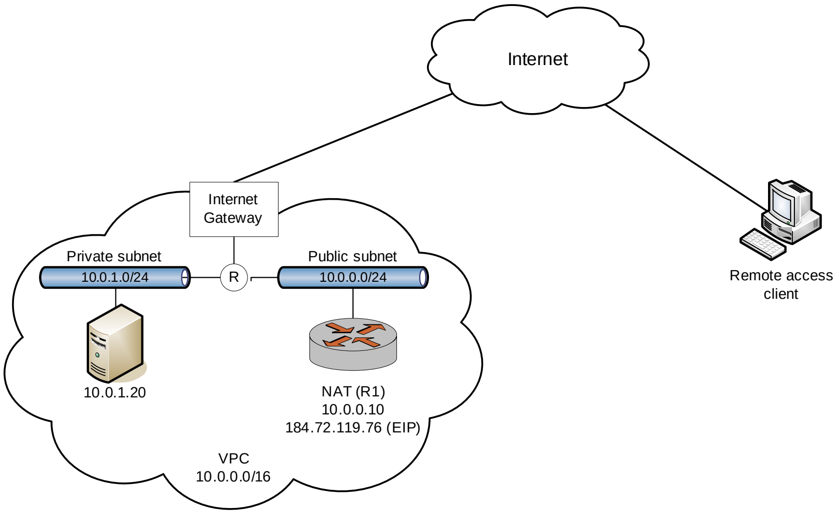
```
vyatta@vyatta:~$ ping 10.0.1.20
PING 10.0.1.20 (10.0.1.20) 56(84) bytes of data.
64 bytes from 10.0.1.20: icmp_req=1 ttl=64 time=0.439 ms
64 bytes from 10.0.1.20: icmp_req=2 ttl=64 time=0.572 ms
64 bytes from 10.0.1.20: icmp_req=3 ttl=64 time=0.430 ms
64 bytes from 10.0.1.20: icmp_req=4 ttl=64 time=0.448 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 38. The following diagram shows the configuration.



To allow inbound Remote Access VPN traffic, add a rule to the default VPC security group. This inbound rule allows TCP traffic on port 1723 from any source (0.0.0.0/0). See "Modify the Default Security Group" on page 41 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 “pptp*X*,” 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 Amazon Web Services.

In this chapter:

Release-Specific Upgrade Information

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

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

Before Upgrading

Before upgrading:

- Save your existing configuration file for reference. Your configuration file is named **config.boot** and is located in the directory **/config**.
- Make sure you have enough space on your root partition to load the image. You can determine the amount of space available using the **show system storage** command.

Upgrading a Vyatta AMI

The Vyatta AMI is supported only for the Vyatta Subscription Edition.

The Vyatta AMI consists of the following:

- The Vyatta virt ISO
- Other AMI-specific modifications and optimizations.

The way you upgrade a Vyatta AMI system depends on what part of the image has changed. Table 3-1 shows the upgrade options for Vyatta AMI.

Table 3-1 Upgrade options for Vyatta AMI systems

What has changed:	**What you need to upgrade:**
The virt ISO	Upgrade just the virt ISO. You can use the upgrade system image command. Use the procedure given in Upgrading the System Image.
AMI-specific modifications	Upgrade the full AMI. Use the procedure given in Upgrading the Full Vyatta AMI
You’re not sure	Use the procedure given in Upgrading the System Image. The system will detect whether anything else in the AMI has changed and will alert you if you need to upgrade the full AMI.

Upgrading the System Image

The **upgrade system image** command provides a simplified, streamlined upgrade process. If the **upgrade system image** command is executed, the system automatically does all of the following:

- Finds the most recent stable Vyatta Subscription Edition virt ISO image
- Downloads the image
- Installs the image
- Migrates configuration files from the running system
- Sets the new image as the default boot image.

The new image is run the next time the system reboots.

To upgrade using the "upgrade system image" command

1. At the command prompt, issue the **upgrade system image** command. Follow the prompts; see the sample session given in Example 3-1.
2. When the install has completed, reboot the system using the **reboot** command. The system restarts using the new system image.

Sample Session for "upgrade system image"

Example 3-1 shows a session where the **upgrade system image** command is used to upgrade to the latest system image.

NOTE *You will not be prompted for your repository username and password if they are already configured within the entitlement system.*

Example 3-1 Upgrading a system image

```
vyatta@vyatta:~$ upgrade system image
Vyatta image upgrade utility.
Please enter repository username: testco
Please enter repository password: testpassword
Checking for updated images on the Vyatta repository...
I have found a newer system image on the Vyatta repository.
The new image is version: VSE6.4-2012.02.09
Would you like to upgrade to this image? [Yes/No] yes
OK...  Starting process to upgrade system image.
Trying to fetch ISO file from
http://packages.vyatta.com/vyatta-supported/iso/stable/vyatta-livecd-vir
t_VSE6.4-2012.02.09_i386.iso
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  196M  100  196M    0     0   489k      0  0:06:49  0:06:49 --:--:--  559k
ISO download succeeded.
```

```
Checking for digital signature file...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   189  100   189    0     0    169      0  0:00:01  0:00:01 --:--:--  2333
Found it.  Checking digital signature...
gpg: directory `/root/.gnupg' created
gpg: new configuration file `/root/.gnupg/gpg.conf' created
gpg: WARNING: options in `/root/.gnupg/gpg.conf' are not yet active during
this run
gpg: keyring `/root/.gnupg/pubring.gpg' created
gpg: Signature made Mon Feb 6 16:42:22 2012 GMT+8 using DSA key ID 9436A9F8
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: Good signature from "Autobuild <autobuild@vyatta.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1B49 FE0A 0239 706A C6D4  13B0 04A2 5B93 9436 A9F8
Digital signature is valid.
Checking MD5 checksums of files on the ISO image...OK.
Done!
What would you like to name this image? [VSE6.4-2012.02.09]: <Enter>
OK.  This image will be named: VSE6.4-2012.02.09
Installing "VSE6.4-2012.02.09" image.
Copying new release files...
Would you like to save the current configuration
directory and config file? (Yes/No) [Yes]: <Enter>
Copying current configuration...
Would you like to save the SSH host keys from your
current configuration? (Yes/No) [Yes]: <Enter>
Copying SSH keys...
Setting up grub configuration...
Done.
```

Upgrading the Full Vyatta AMI

When AMI-specific content in the Vyatta AMI changes, you must perform an upgrade to the new Vyatta AMI, using the procedure in this section.

To upgrade the Vyatta AMI.

1. Save your current system configuration (/config) to a separate location on your network.
2. Using the new Vyatta AMI, create a new Vyatta virtual machine in your AWS environment. Use the instructions given in Chapter 1: Installing the System, starting in the section“Obtaining and Launching the Vyatta AMI” on page 10 .
3. Perform initial configuration of the new virtual machine and test the installation to verify connectivity on the network.

4 Shut down the old system so it does not conflict with the new system.

5 Load the configuration you saved onto the new Vyatta virtual machine.

6 Make the following modification to the loaded configuration:

- For each Ethernet interface, delete the hardware ID. (In configuration mode, use the **delete interface ethernet** ***ethx*** **hw-id** command, where *ethx* is the name of the Ethernet interface).

7 Reboot the system using the **reboot** command. The system restarts using the new configuration.

Chapter 4: Installation and Upgrade Commands

This chapter describes installation and upgrade commands.

This chapter presents the following commands.

Configuration Commands	
None.	
Operational Commands	
add system image	Adds a binary system image to the currently running system.
clone system image	Creates a copy of a Vyatta system image installed on the local system or on a remote system.
delete system image	Deletes a Vyatta system image.
install image	Installs a Vyatta system image, using a binary system image.
install system	Installs Vyatta system software, using a traditional layout of files.
rename system image	Renames a Vyatta system image.
set system image default-boot	Selects a Vyatta system image to be run when the system is next rebooted.
show system image	Displays a list of Vyatta system images installed on the system.
upgrade system image	Upgrades the currently running system to the latest version.

add system image

Adds a binary system image to the currently running system.

Syntax

add system image {*iso-filename* | *iso-URL* [**username** *username* **password** *password*]}

Command Mode

Operational mode.

Parameters

iso-filename	The name of the Vyatta system image file to be added.
iso-URL	The URL location of the Vyatta system image file to be added.
username	Optional. The username required to login to the remote system at the specified URL location.
password	Optional. The password required to login to the remote system at the specified URL location. If the username is specified, then a password must also be specified.

Default

None.

Usage Guidelines

Use this command to add a binary Vyatta system image to the currently running system. A system image can be added to a system that was installed using a disk-based install (using the **install system** command) or an image-based install (using the **install image** command). Once added, it will be set as the new default boot image and will be run the next time the system is booted.

The command will validate the MD5 checksums of the files contained in the ISO image to ensure that it has not been corrupted. In addition, it will not allow more than a single copy of an image to exist on the same system.

The *iso-filename* or *iso-URL* argments provide the source for the ISO image file.

***NOTE** If you are accessing the ISO image on the web, in most browsers right-clicking the link to the file will provide access to the URL which can then be copied and pasted as the iso-URL argument to this command.*

The following table shows the syntax for file specification for different file locations.

Table 4-1

Location	Specification
An absolute path	For *iso-filename* use standard UNIX file specification.
A relative path	For *iso-filename* you can also specify the path name relative to the current directory.
FTP server	Use the following syntax for the *iso-URL* argument: ftp://*user*:*passwd*@*host*/*image-file* where *user* is the username on the host, *passwd* is the password associated with the username, *host* is the host name or IP address of the FTP server, and *image-file* is the ISO image file, including the path. Alternatively, the username and password can be specified as **username** and **password** arguments to the add system image command. If you do not specify *user* and *passwd* you are prompted for them.
SCP server	Use the following syntax for the *iso-URL* argument: scp://*user*:*passwd*@*host*/*image-file* where *user* is the username on the host, *passwd* is the password associated with the username, *host* is the host name or IP address of the SCP server, and *image-file* is the ISO image file, including the path. Alternatively, the username and password can be specified as **username** and **password** arguments to the add system image command. If you do not specify user and passwd you will be prompted for them.
HTTP server	Use the following syntax for the *iso-URL* argument: http://*host*/*image-file* where *host* is the host name or IP address of the HTTP server and *image-file* is the ISO image file, including the path.
TFTP server	Use the following syntax for the *iso-URL* argument: tftp://*host*/*image-file* where *host* is the host name or IP address of the TFTP server, and *image-file* is the ISO image file, including the path relative to the TFTP root directory.

clone system image

Creates a copy of a Vyatta system image installed on the local system or on a remote system.

Syntax

clone system image [*user@host:*]*source-image-name new-image-name* [**clean**]

Availability

Vyatta Subscription Edition.

Command Mode

Operational mode.

Parameters

user	The user name on a remote host. Required for remote host access via SCP. Not required for cloning a local system image.
host	The hostname or IP address of a remote host. Required for remote host access using SCP. Not required for cloning a local system image.
source_image-name	The name of the system image to be copied. The source image can exist on the local system or a remote system.
new-image-name	The name of the new (copied) system image. An image with this name must not already exist on the system.
clean	Creates an empty read-write directory tree for the new image. This creates a new image that is functionally equivalent to the source image as it existed when it was originally installed.

Default

None.

Usage Guidelines

Use this command to create a copy of a system image installed on the local system or on a remote system to the local system.

If *user@host* is specified, the image is fetched from the named host using the SCP protocol. If *user@host* is omitted, the *source-image-name* is the name of an image that already exists on the system. The *new-image-name* is the image name that the system uses for the clone. There must be no image by that name already existing on the system.

Command completion is performed for local image names if *user@host* is not specified. No command completion is performed on remote image names if *user@host* is specified.

If the **clean** argument is omitted, the command copies the **squashfs** file being used by the image named *source-image-name* as well as the read-write directory tree of *source-image-name*. If the **clean** argument is given, then the read-write directory tree of *source-image-name* is NOT copied. Instead, an empty read-write directory tree is created for the new image. This creates a new image that is functionally equivalent to the source image as it existed when it was initially installed.

Images created by this command behave the same as images installed by the **install image** or the **add system image** commands.

The **https** and **ssh** services must both be enabled on the remote Vyatta system in order for the clone system image command to work properly. The **https** service is enabled using **set service https** in Configuration mode. The **ssh** service is enabled using **set service ssh** in Configuration mode.

NOTE *This command is only available in the Vyatta Subscription Edition.*

delete system image

Deletes a Vyatta system image.

Syntax

delete system image [*image-name*]

Command Mode

Operational mode.

Parameters

image-name	The name of the Vyatta system image to be deleted.

Default

When used with no options, the system prompts for the image to delete.

Usage Guidelines

Use this command to delete a Vyatta system image from the local disk drive.

The image and all of its local files, including its Vyatta configuration file, are all destroyed. Since this command is destructive, the system prompts for confirmation.

Command completion displays all valid completions for the *image-name* argument. If the *image-name* argument is omitted, the system displays a list of available images and prompts you to select one.

If the system was originally installed in disk-based mode, an **image-name** option is available that you can use to direct that the disk-based installation should be deleted.

The system does not allow you to delete the currently running system image. However, the system does allow you to delete the image currently selected to be run at the next reboot. If you choose this, the system uses the currently running image when the system is next rebooted.

install image

Installs a Vyatta system image, using a binary system image.

Syntax

install image

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to install a Vyatta system binary image.

This command is similar to the **install system** command in functionality. Once the installation is complete you can add multiple Vyatta versions into the same partition, using the **add system image** command, and you can then choose which version to boot, using the **set system image default-boot** command. This allows you to move easily between different versions of the system.

If you have a new system and want to install the Vyatta system from scratch, you can boot the Vyatta LiveCD and then run the **install image** command to install the image on the LiveCD to the disk. The **install image** command operates similarly to the **install system** command—it creates and formats a new disk partition and then installs the image to the partition while preserving the system configuration.

install system

Installs Vyatta system software, using a traditional layout of files.

Syntax

install system

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to install Vyatta software from a LiveCD onto a persistent device such as a hard disk.

NOTE *Vyatta recommends using the **install image** command over the **install system** command.*

If you have a new system and want to install the Vyatta system from scratch, you can boot the Vyatta LiveCD and then run the **install system** command to install the system on the LiveCD to the disk. The **install system** command operates similarly to the **install image** command—it creates and formats a new disk partition and then installs the system to the partition while preserving the system configuration.

rename system image

Renames a Vyatta system image.

Syntax

rename system image *old-image-name new-image-name*

Command Mode

Operational mode.

Parameters

old-image-name	The name of an existing Vyatta system image to be renamed.
new-image-name	The new name of the Vyatta system image.

Default

None.

Usage Guidelines

Use this command to rename a Vyatta system image.

The old name must match the name of an image on the system. The system does not allow you to rename the currently running system image. The new system image name cannot be in use by another image.

set system image default-boot

Selects a Vyatta system image to be run when the system is next rebooted.

Syntax

set system image default-boot [*image-name*]

Command Mode

Operational mode.

Parameters

image-name	The name of the Vyatta system image to be run when the system is rebooted.

Default

If used with no image name specified, the system displays a list of available images and prompts you to select one.

Usage Guidelines

Use this command to specify which Vyatta system image is to be run when the system is next rebooted.

When multiple system images have been installed using the **add system image** command, you can use this command to direct the system to boot from a specific system image the next time it is restarted.

Command completion displays all valid completions for the *image-name* argument. If the *image-name* argument is omitted, the system displays a list showing all images installed on the system and prompts you to select one. If the system was originally installed in disk-based mode, then a special **image-name** option is available so that you can select the disk-based system as the default system from which to boot.

show system image

Displays a list of Vyatta system images installed on the system.

Syntax

show system image [storage | version]

Command Mode

Operational mode.

Parameters

storage	Display the amount of disk space used by each image.
version	Include the image version number in the display of system images.

Default

None.

Usage Guidelines

Use this command to display a list of all Vyatta system images currently installed on the system.

The command output identifies the image that is currently running, as well as the image that has been selected to run when the system is next rebooted. If the system was originally installed in disk-based mode, then one of the image names identifies that installation.

upgrade system image

Upgrades the currently running system to the latest version.

Syntax

upgrade system image

Availability

SE

Vyatta Subscription Edition.

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to upgrade the Vyatta system image to the latest release. It is the preferred method of system upgrade. The system image can be upgraded on a system that was installed using a disk-based install (using the **install system** command) or an image-based install (using the **install image** command or from a virtual machine template). Once the new image is added to the system, the configuration from the currently running system can be migrated. Also, the new image will be set as the new default boot image and will be run the next time the system is booted.

The command will validate the MD5 checksums of the files contained in the ISO image to ensure that it has not been corrupted. In addition, it will not allow more than a single copy of an image to exist on the same system.

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/Ouput
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
PTMU	Path Maximum Transfer Unit

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
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
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