VI. **Step 6.0 – FortiGate Configuration** ............................................................................................................31
   Step 6.1 - Update FortiGate Password
   Step 6.2 – Confirm network settings
   Step 6.3 – Setup basic policies

VII. **Step 7 – Testing** ...........................................................................................................................................34
   Step 7.1 – Launch a RDP session to test
   Step 7.2 – Retrieve your VM’s password
   Step 7.3 – Test Outbound access

VIII. **Appendix** ....................................................................................................................................................37
   Regions and Availability Zones
   Amazon EC2 Key Pairs
   Additional info and links

*Change History*

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author.</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>8-1-2014</td>
<td>Justin L. Wireman</td>
<td>Initial Document creation</td>
</tr>
</tbody>
</table>
Overview

This document is design to be a quick start walk-through in setting up a virtual Fortinet device utilizing the AWS services. We will start out reviewing some of the AWS concepts.

Amazon Virtual Private Cloud (Amazon VPC)

Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can also create a Hardware Virtual Private Network (VPN) connection between your corporate datacenter and your VPC and leverage the AWS cloud as an extension of your corporate datacenter.

You can easily customize the network configuration for your Amazon VPC. For example, you can create a public-facing subnet for your web servers that have access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.
Components of Amazon VPC

Amazon VPC is comprised of a variety of objects that will be familiar to customers with existing networks:

- **A Virtual Private Cloud (VPC)**: a logically isolated virtual network in the AWS cloud. You define a VPC’s IP address space from a range you select.
- **Subnet**: a segment of a VPC’s IP address range where you can place groups of isolated resources.
- **Internet Gateway**: the Amazon VPC side of a connection to the public Internet.
- **NAT Instance**: An EC2 instance that provides Port Address Translation for non-EIP instances to access the Internet via the Internet Gateway.
- **Hardware VPN Connection**: a hardware-based VPN connection between your Amazon VPC and your datacenter, home network, or co-location facility.
- **Virtual Private Gateway**: the Amazon VPC side of a VPN Connection.
- **Customer Gateway**: Your side of a VPN Connection.
- **Router**: Routers interconnect Subnets and direct traffic between Internet Gateways, Virtual Private Gateways, NAT instances and Subnets.
- **Peering Connection**: A peering connection enables you to route traffic via private IP addresses between two peered VPCs.

How do instances in a VPC access the Internet?

Elastic IP addresses (EIPs) give instances in the VPC the ability to both directly communicate outbound to the Internet and to receive unsolicited inbound traffic from the Internet (e.g., web servers)

How do instances without EIPs access the Internet?

Instances without EIPs can access the Internet in one of two ways:

Instances without EIPs can route their traffic through a NAT instance to access the Internet. These instances use the EIP of the NAT instance to traverse the Internet. The NAT instance allows outbound communication but doesn’t enable machines on the Internet to initiate a connection to the privately addressed machines using NAT, and

For VPCs with a Hardware VPN connection, instances can route their Internet traffic down the Virtual Private Gateway to your existing datacenter. From there, it can access the Internet via your existing egress points and network security/monitoring devices.
Network Information

Figure #2 the default network design for a Public and Private VPC. We will be replacing much of the Router functionality with the FortiGate as pictured in figure 1.

- VPC Subnet – 10.0.0.0/16
- Public Subnet - 10.0.0.0/24
- Private Subnet – 10.0.1.0/24

Video Walk-through

A video version of this guide will be available on SE Resource Site
Basic AWS Network Setup

Step 1 – Setting up your AWS account

You will need to provide billing information to setup an AWS account. Once you have completed the basic account setup you will be presented with the AWS console.

For more information on AWS check out the getting started guide. Click here

Figure 3
Step 2 – Create a Virtual Private Cloud (VPC)

To allow VM instances access to more than one interface you need to create a VPC (virtual private cloud). You need to change dashboards to VPC and for our purpose start the VPC wizard.

It is important to note that like most multi-tenant environments AWS reserves the first 5 IP address of each network that is created for its own router / firewall and DHCP / DNS servers.
Step 2.1 – VPC Wizard

This next section is a visual walk-through of the VPC wizard. Select the Public and Private subnet option.

One item to double check on step 2 of the VPC wizard is to make sure that both subnets are in the same availability zone. Please see the Appendix for more information on availability zones.
Once you have verified the network setting, click create VPC and you will see the screen below.

![VPC Wizard](image1)

Figure 7 - VPC Wizard

When the VPC setup has been completed you can review subnet and routing information on the VPC Dashboard. More on this later in the guide, as you will need to alter settings to route traffic through the FortiGate.

![VPC Dashboard](image2)

Figure 8
Step 3 – EC2 Launching virtual machines

Change dashboards to the EC2 dashboard. For time sake it is normally faster to get the VM provisioning started while setting up the network. Click Launch Instance on this screen.

![Figure 9 - EC2 Dashboard](image-url)
Step 3.1 – Choosing an AMI

For this guide we have chosen the Bring your Own License version of the FortiGate VM.
Step 3.2 – Instance type

Choose the instance type that matches the license. For this example I have a 1 vCPU license file.
Step 3.3 – Instance Details

In this step you will choose the public subnet, assign IP addresses, and add the eth1 interface (private subnet).
Step 3.4 – Instance Storage

If you are configuring this for demonstration purposes, you can change the highlighted storage size to create a larger disk size for logging / reporting.

![Figure 12 - Step 4: Add Storage](image)

Step 3.5 – Instance Tags

It is valuable to create tags to quickly reference instance items in your AWS deployment. I have tagged a few items below as an example.

![Figure 13 - Instance Tags](image)
Step 3.6 – Security groups

Amazon by default has your VPC behind a basic firewall. Since we are going to be utilizing the FortiGate, I have created a Permit All security group and applied it to this instance.

---

Step 3.7 – Key Pair and Launch Instance

- Choose proceed without a keypair and use the default FortiGate username / password.
- Click Launch Instance to begin the provisioning.

---
In this section you will be locating items such as the Network interface ENI on the EC2 dashboard and making IP and routing updates on the VPC dashboard.

Step 4 – Configure AWS network settings

Step 4.1 - Associate a public “elastic” IP to the FG-VM public interface

- On the EC2 Dashboard under the Network interface menu.
  - Locate the public interface ENI.
    - See step 4.3 figure 18 for a screenshot of this menu.
- On the VPC Dashboard under the Elastic IPs menu.
  - If the Public IP is associated with a default instance you will need to disassociate the Public IP before you can proceed.
  - Use the ENI of the public FortiGate interface as the object to associate the public IP.
Figure 16
Step 4.2 – Confirm the assigned Public address

- Take note of the public IP address and DNS assigned. You will use these items in later steps.
Step 4.3 – Setting up the default route for the private network.

- On the EC2 Dashboard under the Network interface menu.
  - Locate the network interface ID (ENI-) of the private network and Copy the ID.
- Change dashboards back to the VPC>Route Tables
  - Edit the default route (for the private subnet) to point to the FortiGate private network interface ID.
  - Demonstrated in figures 19-20

![Figure 18](image18.jpg)

![Figure 19](image19.jpg)
- Associate the private subnet to the private routing entry you have been editing in the previous steps.

![Figure 20](image1)

![Figure 21](image2)

- On the EC2 Dashboard under the Network interface menu.
  - Right click and select Change Source/Dest Check
  - Select Disable and Save
Step 4.5 - Navigate to EC2 dash to review the Instance state

- Once confirming that the instance has finished provisioning and powering up check the following items.
  - Public IP/DNS assigned
  - Confirm the correct security group is assigned.
Step 4.6 - Access the Virtual FortiGate

- Open a HTTPS session to the public IP or DNS entry provided and login with the default username / password.
- Upload license file for BYOL. (See figure 22)
- The FortiGate will reboot after license install.

![Figure 22](image)

Step 4.7 – SSH to the FortiGate

- SSH to the device using the DNS hostname
- Issue the following commands to test access
  - Ping 8.8.8.8 to test connectivity
  - Execute update-now
  - Execute formatlogdisk and reboot (Option step if you need disk logging)

```
FortiGate-VM64-AWS# Execute ping 8.8.8.8
FortiGate-VM64-AWS# Execute update-now
FortiGate-VM64-AWS# Execute formatlogdisk
```
Step 5.0 – Setup a Test VM

In this step we will setup a test windows VM on the private network and configure it to use the FortiGate for all access in and out of the private network. In this example we are going to setup remote access to the test VM as well.

Step 5.1 – Provision a new AMI

For this example we are using Windows Server 2012. Note that any OS version could be used for testing / demonstration purposes.
Step 5.2 – Select a VM Instance type

The default is the free tier general purpose type. This instance type is fine for basic testing.
Step 5.3 – Choose Instance settings

- It is important to select the private subnet to place this VM behind the FortiGate.
- I have also chosen to assign the IP address of 10.0.1.25. I have done this so I can setup port forwarding on the FortiGate while this VM is provisioned.

![Figure 25](image-url)
Step 5.4 – VM Storage settings
Adjust the default storage setting as appropriate for our virtual machine. If you are deploying this machine for basic testing the default should suffice.

Step 5.5 – Assign any tags needed to the VM Instance
This is an optional step.

Step 5.6 – VM Security Group Settings
Assign the same “Permit All” security group you created during the step 3.6.
Step 5.7 – Review Instance Settings and Launch Instance

Figure 28
Step 5.8 – Create key pair

If you already have a key pair you can use an existing one. If not choose to create a new key pair and download it. You will need this file to login to the VM.

**Important** - If you lose the key pair, you cannot connect to your Amazon EC2 instances.

For more information on Key Pairs see the Appendix.
Step 6.0 – FortiGate Configuration

While the Windows VM is being provisioned you can finish the FortiGate configuration.

Step 6.1 - Update FortiGate Password

Update the FortiGate password as there are many bots that attempt to log in to newly provisioned devices on AWS subnets.
Step 6.2 – Confirm network settings
Set the port2 interface IP address settings (private subnet)

Step 6.3 – Setup basic policies
For this example we are going to create the following policies. (Samples below)

- NAT & allow outbound access
  - (Optional) You can apply any additional policies if you want to demonstrate features such as Web-filtering, DLP, etc.
- Port forwarding port 3389 to the Windows server
- Any required logging for troubleshooting

```diff
config firewall policy
edit 1
set srcintf "port2"
set dstintf "port1"
set srcaddr "all"
set dstaddr "all"
set action accept
set schedule "always"
set service "ALL"
set utm-status enable
set logtraffic all
set av-profile "default"
set ips.sensor "default"
set profile-protocol-options "default"
set nat enable
next
```
edit 2
   set srcintf "port1"
   set dstintf "port2"
   set srcaddr "all"
   set dstaddr "Windows-RDP"
   set action accept
   set schedule "always"
   set service "ALL"
   set utm-status enable
   set logtraffic all
   set av-profile "AV-flow"
   set ips-sensor "default"
   set profile-protocol-options "default"
   next
end

config firewall vip
   edit "Windows-RDP"
   set extintf "port1"
   set portforward enable
   set mappedip 10.0.1.25
   set extport 3389
   set mappedport 3389
   next
end
Step 7 – Testing

Step 7.1 – Launch a RDP session to test

Figure 31

Step 7.2 – Retrieve your VM’s password

On the EC2 Dashboard, Right click your test VM instance and select Get Windows Password

Figure 32
• You will be asked for the key pair you created to decrypt the administrator password.
Step 7.3 – Test Outbound access

For testing purposes I have attempted to download a file from eicar.org to show that the FortiGate is inline for outbound traffic. See Figures 34-35.
Regions and Availability Zones

Region and Availability Zone Concepts

Each region is completely independent. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links. The following diagram illustrates the relationship between regions and Availability Zones.

You can list the Availability Zones that are available to your account. For more information, see Describing Your Regions and Availability Zones. When you launch an instance, you can select an Availability Zone or let us choose one for you. If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in another Availability Zone can handle requests.

Amazon EC2 resources are either global, tied to a region, or tied to an Availability Zone. For more information, see AWS documentation for the complete article.
Amazon EC2 Key Pairs

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a key pair.

To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance. Linux/Unix instances have no password, and you use a key pair to log in using SSH. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP.

Creating a Key Pair

You can use Amazon EC2 to create your key pair. For more information, see Creating Your Key Pair Using Amazon EC2. Alternatively, you could use a third-party tool and then import the public key to Amazon EC2. For more information, see Importing Your Own Key Pair to Amazon EC2.

Each key pair requires a name. Be sure to choose a name that is easy to remember. Amazon EC2 associates the public key with the name that you specify as the key name. Amazon EC2 stores the public key only, and you store the private key. Anyone who possesses your private key can decrypt your login information, so it's important that you store your private keys in a secure place.

The keys that Amazon EC2 uses are 1024-bit SSH-2 RSA keys. You can have up to five thousand key pairs per region.

Launching and Connecting to Your Instance

When you launch an instance, you should specify the name of the key pair you plan to use to connect to the instance. If you don't specify the name of an existing key pair when you launch an instance, you won't be able to connect to the instance. When you connect to the instance, you must specify the private key that corresponds to the key pair you specified when you launched the instance. Amazon EC2 doesn't keep a copy of your private key; therefore, if you lose your private key, there is no way to recover it. If you lose the private key for an instance store-backed instance, you can't access the instance; you should terminate the instance and launch another instance using a new key pair. If you lose the private key for an EBS-backed instance, you can regain access to your instance. For more information, see Connecting to Your Instance if You Lose Your Private Key.
Detailed VPC Diagram

Additional info and links

http://aws.amazon.com/documentation/vpc/