# FortiGate Firewall Deployment on AWS

**By: Ritvik Indupuri**

## 1. Introduction

This document details the deployment and configuration of a FortiGate Next-Generation Firewall (NGFW) within an Amazon Web Services (AWS) cloud environment. The primary objective of this project was to establish a secure, controlled network architecture, demonstrating key cloud networking principles, firewall policy enforcement, and Unified Threat Management (UTM) capabilities, specifically web filtering. The solution isolates a Windows test virtual machine (VM) in a private subnet, with all its inbound and outbound traffic routed and inspected by the FortiGate firewall.

## 2. Architecture Overview

The deployed architecture leverages AWS Virtual Private Cloud (VPC) to create an isolated network environment. A FortiGate firewall instance serves as the central security gateway, positioned between public internet access and a private subnet hosting a Windows test VM. All traffic destined for the Windows VM from the internet, and all outbound traffic from the Windows VM to the internet, traverses the FortiGate for inspection and policy enforcement.

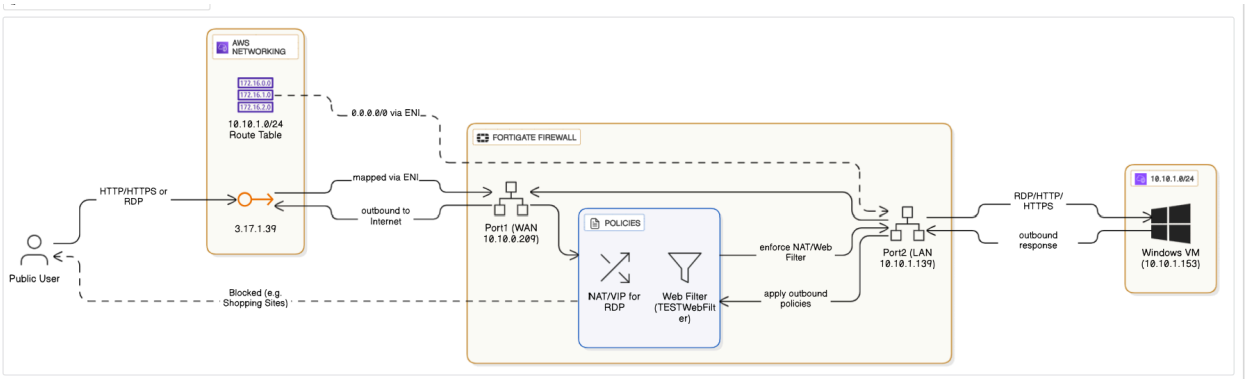

*Figure 2.1: Project Architecture Diagram*

The figure shows a FortiGate firewall deployment on AWS, detailing the logical traffic flow and key components. It illustrates how a public user accesses a Windows VM via the FortiGate's public IP and a Virtual IP (VIP), and how the FortiGate's web filter screens outbound traffic from the Windows VM.

---

## 3. Deployment Steps

The deployment process involved six major steps: VPC and subnet creation, Internet Gateway configuration, FortiGate VM deployment, Windows Test VM deployment, necessary component configuration (route tables, FortiGate interfaces, VIPs, policies), and testing of UTM filters.

### 3.1 AWS VPC and Subnet Creation

A new Virtual Private Cloud (VPC) was created to establish a logically isolated network.

- **VPC Name:** FortiVPC
- IPv4 CIDR Block: 10.10.0.0/23
  Within this VPC, two subnets were provisioned:
- **Public Subnet (PublicSN):** 10.10.0.0/24 - Designated for public-facing resources, including the FortiGate's external interface.
- **Private Subnet (PrivateSN):** 10.10.1.0/24 - Designated for internal resources, hosting the Windows test VM.

### 3.2 Internet Gateway Configuration

An Internet Gateway (IGW) was created and attached to the FortiVPC. This component enables communication between the VPC and the internet, facilitating both inbound and outbound internet access for resources within the public subnet.

- **Internet Gateway ID:** igw-04eb2bdb2c32c796 (example ID)

### 3.3 FortiGate VM Deployment and Initial Setup

A FortiGate-VM64-AWS instance was deployed from the AWS Marketplace. This instance serves as the NGFW for the environment.

- **Instance Type:** t3.small (selected for demo purposes)
- **Network Interfaces:** The FortiGate VM was configured with two Network Interfaces (ENIs):
  - **Public Interface (Port1):** Placed in PublicSN, assigned a private IP (10.10.0.209), and associated with an Elastic IP (3.17.1.39) for public accessibility.
  - **Private Interface (Port2):** Placed in PrivateSN, assigned a private IP (10.10.1.139), acting as the gateway for the Windows VM.

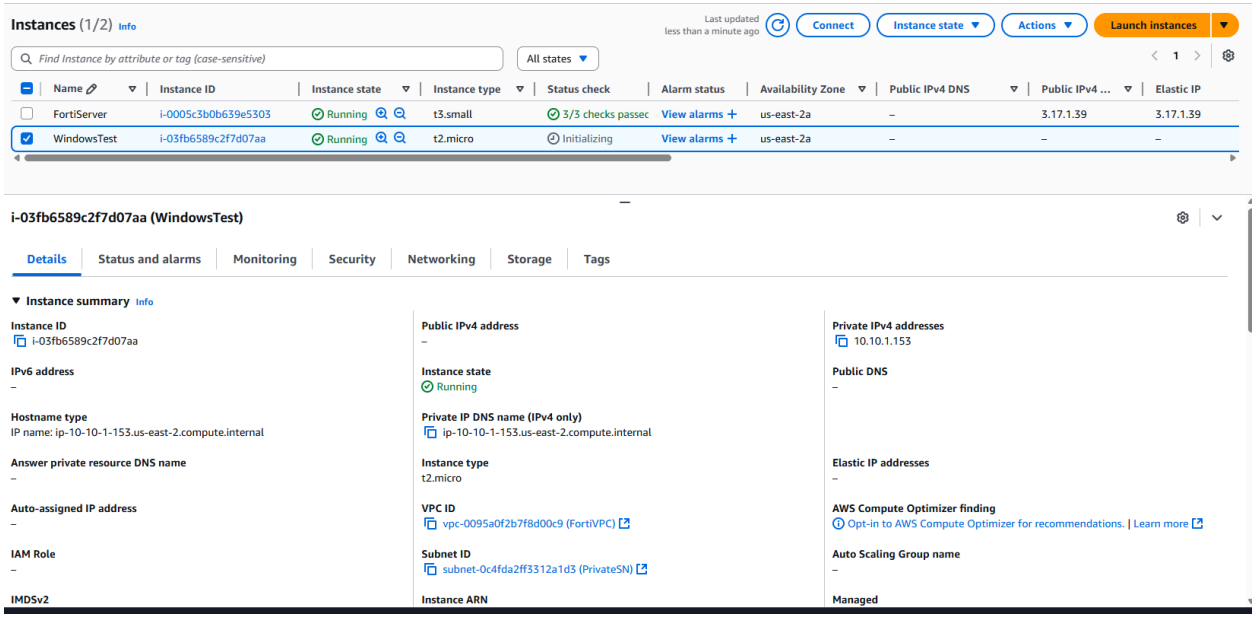


*Figure 3.3.1: AWS EC2 Instances Overview*

The figure shows both the FortiGate firewall (FortiServer) and the Windows test VM (WindowsTest) confirmed as operational, appearing as running EC2 instances in the AWS console.

### 3.4 Windows Test VM Deployment

A Windows Server 2016 VM was deployed to serve as the internal test client.

- **Instance Type:** t2.micro (free tier)
- **Subnet:** PrivateSN (10.10.1.0/24)
- Private IP Address: 10.10.1.153
  Crucially, this VM was deployed without any public IP address, ensuring that all its internet-bound traffic is forced through the FortiGate firewall.

### 3.5 AWS Route Table Configuration

Two distinct route tables were configured to manage traffic flow within the VPC:

- **Public Subnet Route Table (PublicSNRT):**
  - Associated with PublicSN.
  - Contains a default route (0.0.0.0/0) pointing to the Internet Gateway, allowing public internet access.

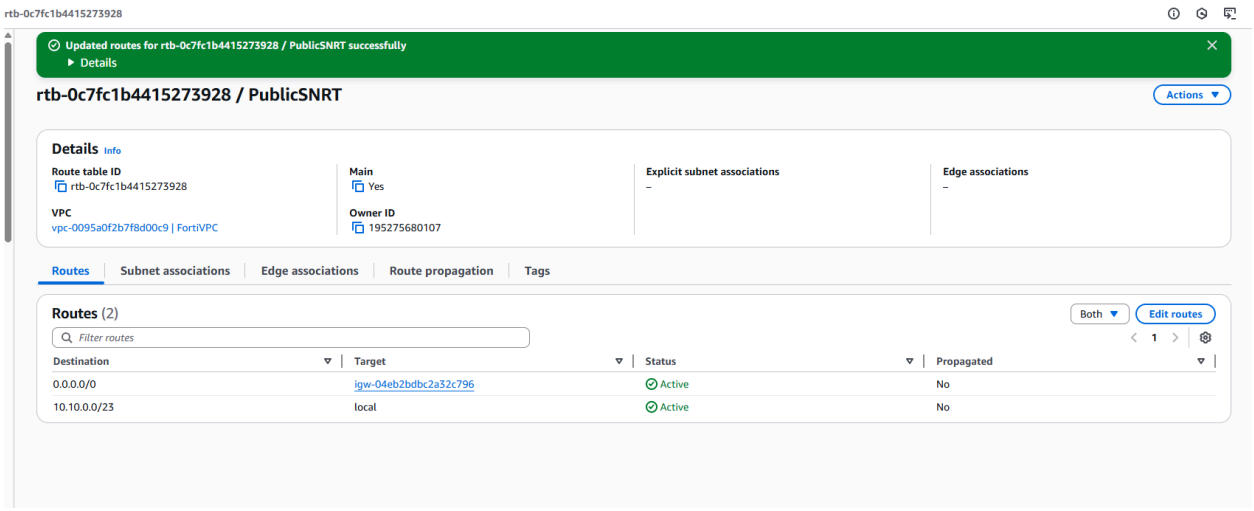


*Figure 3.5.1: Public Subnet Route Table*

This image illustrates the route table for the public subnet. It specifically highlights the default route (0.0.0.0/0), which is configured to direct traffic to the Internet Gateway (igw-...). This configuration enables resources within this subnet to establish outbound internet connectivity.

- **Private Subnet Route Table (PVTSNRT):**
  - Associated with PrivateSN.
  - Contains a default route (0.0.0.0/0) pointing to the private network interface (eni-08c21834ef048c233) of the FortiGate firewall. This ensures all outbound traffic from the private subnet is routed through the FortiGate.

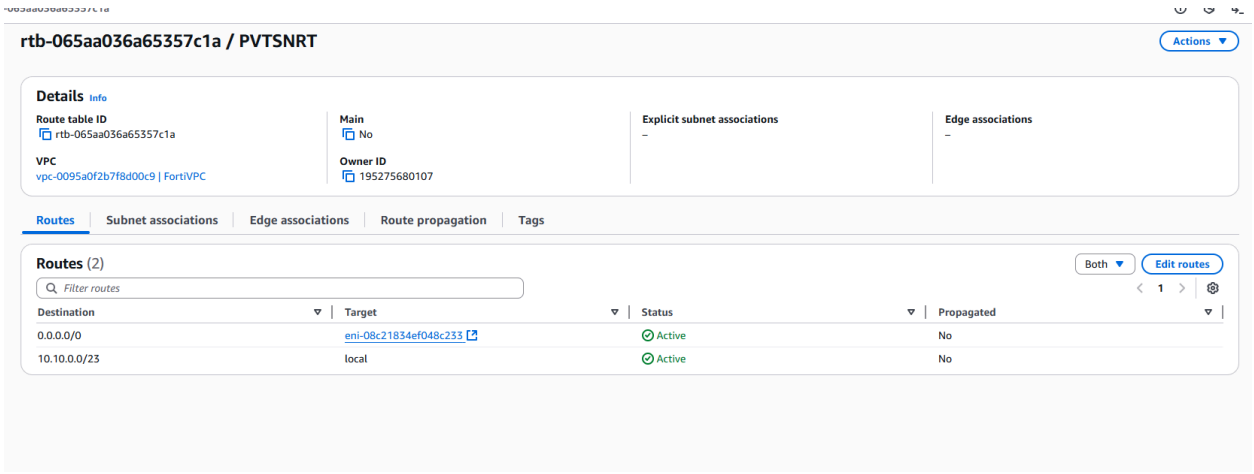


*Figure 3.5.2: Private Subnet Route Table*

The figure shows the private subnet's route table. It clearly shows that all traffic from the private subnet is routed through the FortiGate firewall, as indicated by the default route (0.0.0.0/0) pointing to FortiGate's private network interface.

### 3.6 Disabling Source/Destination Check

For the FortiGate instance to function correctly as a network appliance (i.e., to forward traffic not explicitly addressed to itself), the "Source / destination check" was disabled on both of its network interfaces.

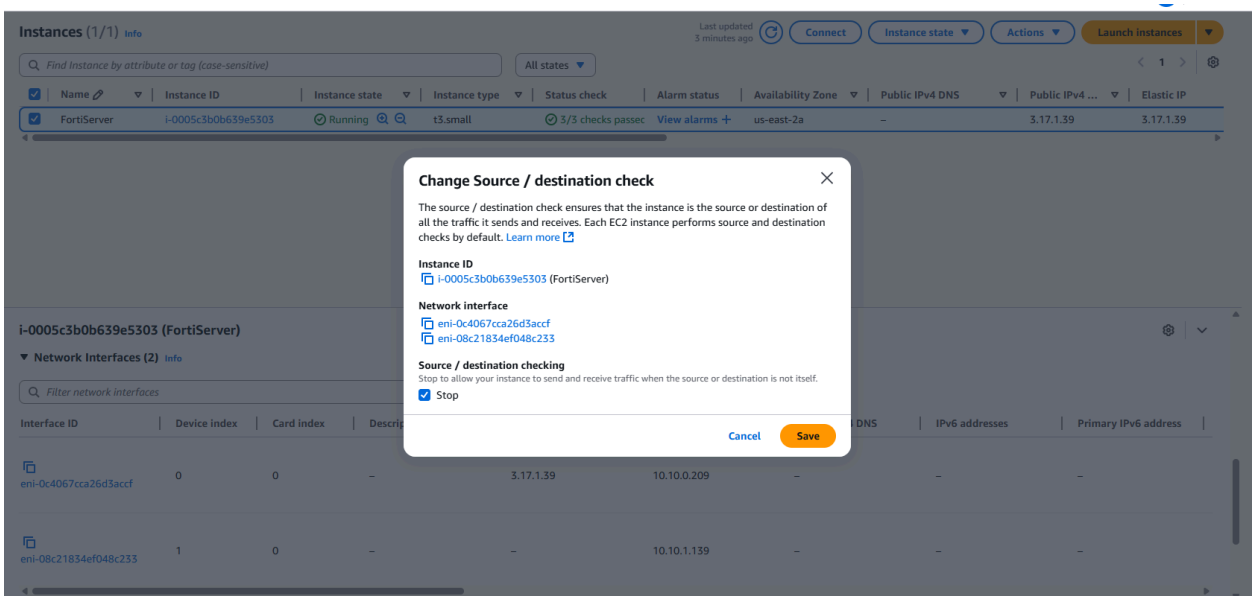


*Figure 3.6.1: Source / Destination Check Disabled*

The figure shows a confirmation prompt for disabling the "Source / destination check" on a FortiGate network interface. This is a crucial step for the FortiGate to act as a gateway, routing traffic for other instances.

## 3.7 FortiGate Interface Configuration

Upon initial login to the FortiGate management console, the two network interfaces (Port1 and Port2) were recognized and configured with their respective IP addresses and administrative access settings.

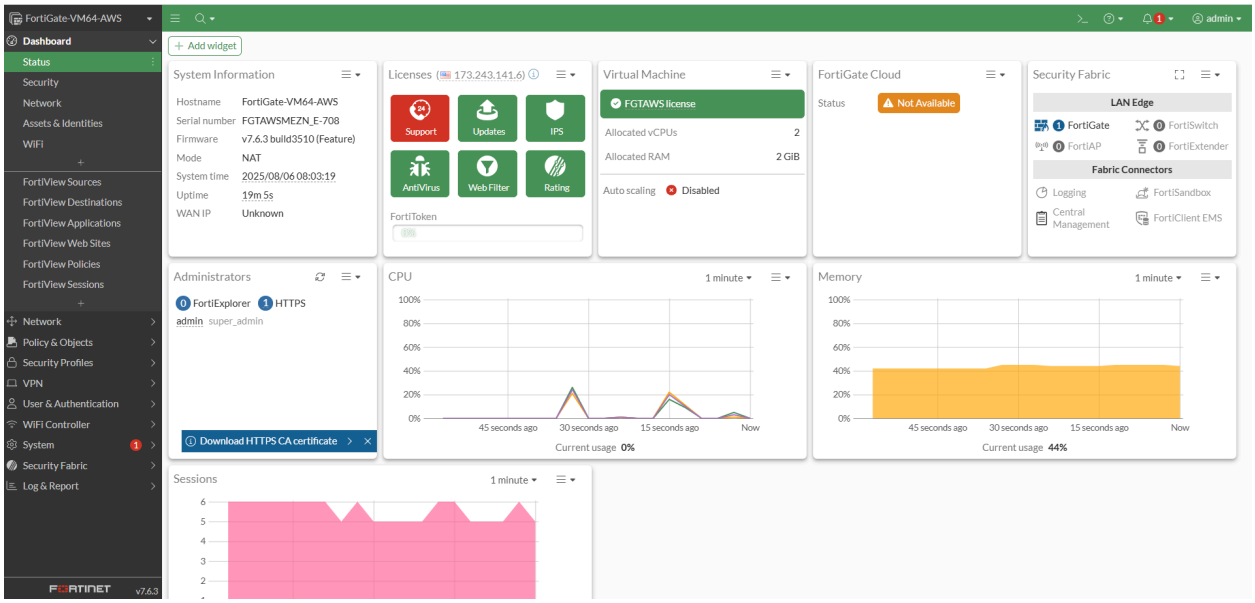


*Figure 3.7.1: FortiGate Dashboard*

The figure shows that the FortiGate dashboard provides a comprehensive overview of the firewall's status, including system details, licensing, resource utilization (CPU and memory), and the operational state of security features like AntiVirus and Web Filtering.

## 3.8 FortiGate Virtual IP (VIP) Creation

A Virtual IP (VIP) was configured on the FortiGate to allow external RDP access to the internal Windows VM.

- **VIP Name:** Windows VIP
- **Interface:** port1 (Public-facing interface)
- **External IP Address/Range:** 0.0.0.0 (any external source)
- **Mapped to IPv4 Address/Range:** 10.10.1.153 (Windows VM private IP)

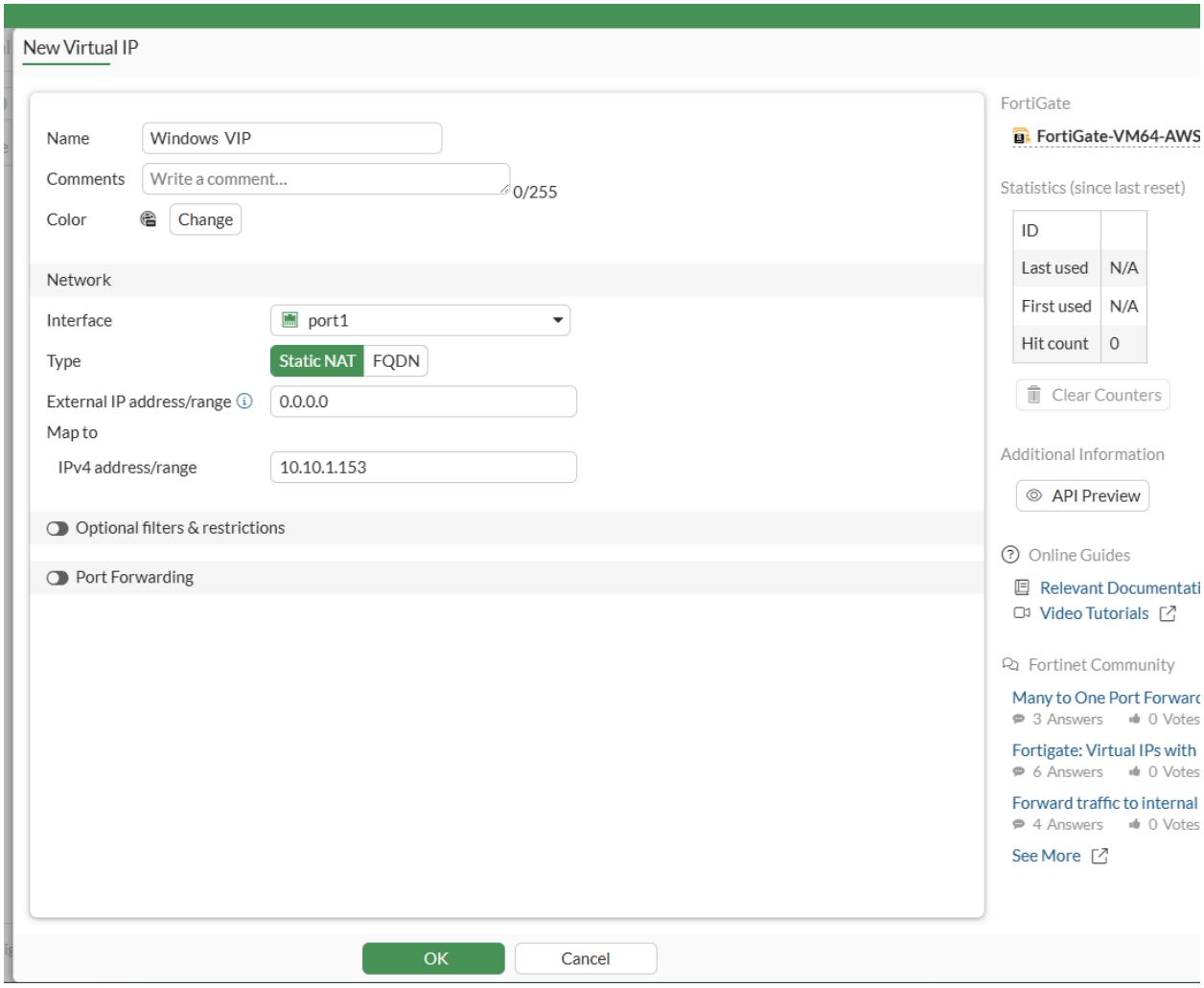


*Figure 3.8.1: FortiGate Virtual IP Configuration*

The figure shows the FortiGate Virtual IP (VIP) configuration, which maps external traffic from port1 to the Windows VM's internal IP address (10.10.1.153) to enable inbound access.

### 3.9 FortiGate Firewall Policies

Two primary firewall policies were created to control traffic flow:

- **WAN to LAN (Inbound RDP):**
  - **Incoming Interface:** port1 (WAN)
  - **Outgoing Interface:** port2 (LAN)
  - **Source:** all
  - **Destination:** Windows VIP
  - **Service:** RDP
  - **Action:** ACCEPT
  - **NAT:** Enabled

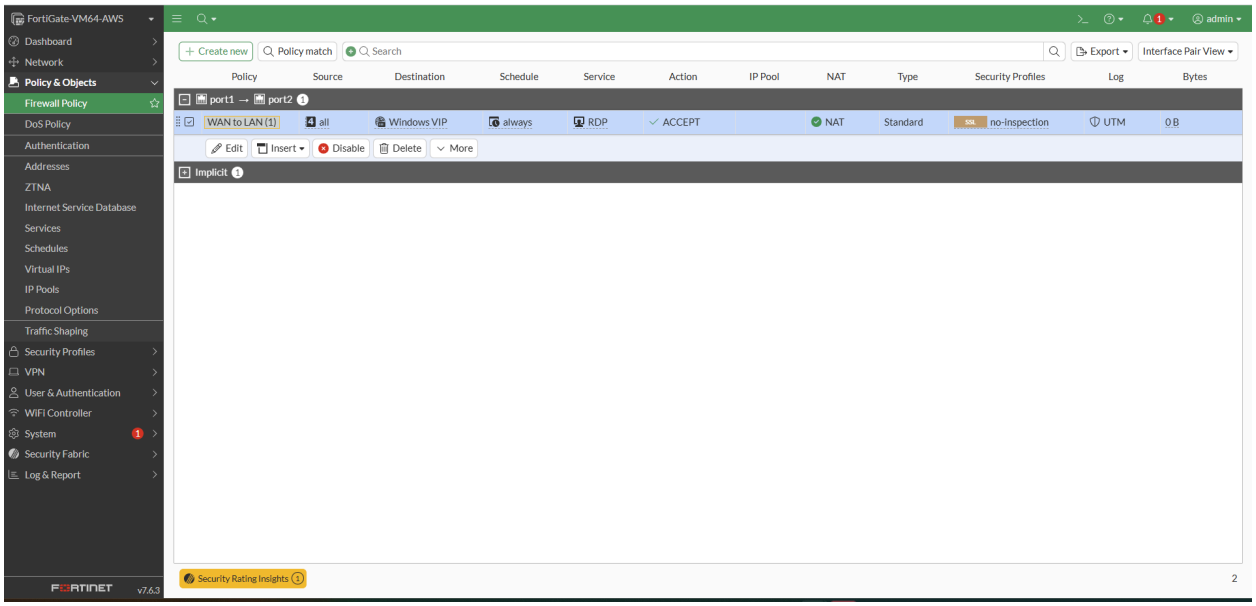


*Figure 3.9.1: FortiGate WAN to LAN Policy*

The figure shows the FortiGate firewall policy, which allows inbound RDP traffic to the Windows VM from any source. This is done by using a configured Virtual IP with NAT enabled.

- **LAN to WAN (Outbound Web Access with UTM):**
  - **Incoming Interface:** port2 (LAN)
  - **Outgoing Interface:** port1 (WAN)
  - **Source:** WindowsVM (Address object for 10.10.1.153)
  - **Destination:** all
  - **Service:** HTTP, HTTPS
  - **Action:** ACCEPT
  - **NAT:** Enabled
  - **Security Profiles:** TESTWebFilter (Web Filter) applied.

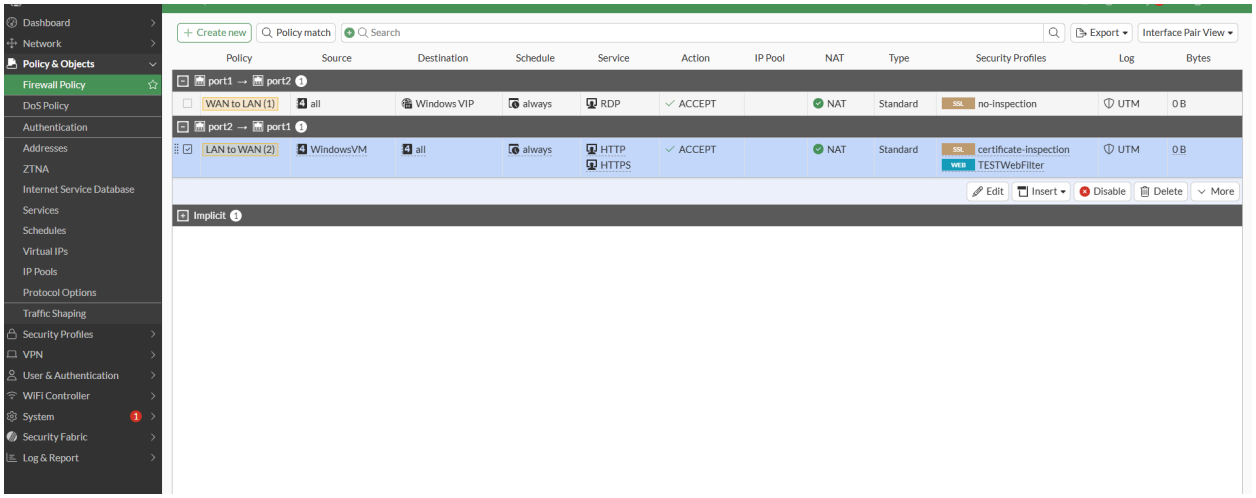


*Figure 3.9.2: FortiGate LAN to WAN Policy with Web Filter*

The figure shows the FortiGate firewall policy, which facilitates outbound HTTP/HTTPS traffic from the Windows VM to the internet. This policy incorporates the TESTWebFilter security profile for content inspection.

### 3.10 FortiGate Web Filter Configuration

A custom web filter profile, TESTWebFilter, was created to demonstrate FortiGate's Unified Threat Management (UTM) capabilities.

- **Name:** TESTWebFilter
- **Feature Set:** Flow-based or Proxy-based
- **FortiGuard Category Based Filter:** Configured to **Block** the "Shopping" category.

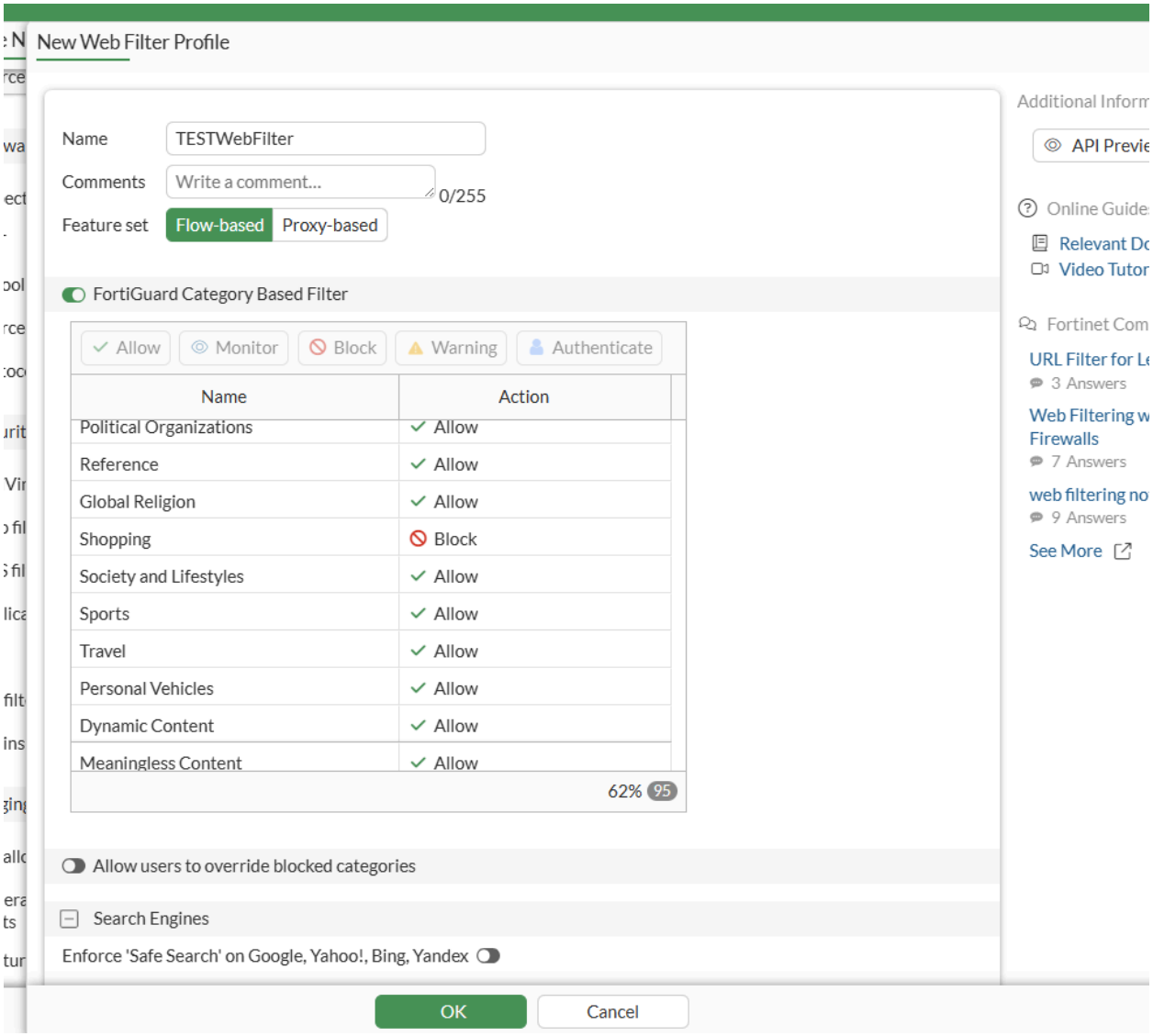


*Figure 3.10.1: FortiGate Web Filter Profile*

The figure shows that FortiGate's TESTWebFilter profile is configured to block the "Shopping" content category.

### 3.11 FortiGate Static Route

A static route was configured on the FortiGate to direct all outbound internet traffic from the private network.

- **Destination:** 0.0.0.0/0 (Default route)
- **Interface:** port1 (Public-facing interface)
- **Gateway:** 10.10.0.66 (The default gateway of the public subnet, which is the AWS-provided router for 10.10.0.0/24)

---

## 4. Testing and Validation

Comprehensive testing was performed to validate the connectivity, firewall policies, and web filtering functionality.

### 4.1 RDP Access to Windows VM

Remote Desktop Protocol (RDP) access to the Windows VM was successfully established using FortiGate's public Elastic IP and the configured Virtual IP. This confirmed that the inbound NAT and firewall policy were functioning correctly.

### 4.2 Internet Connectivity Test (Ping)

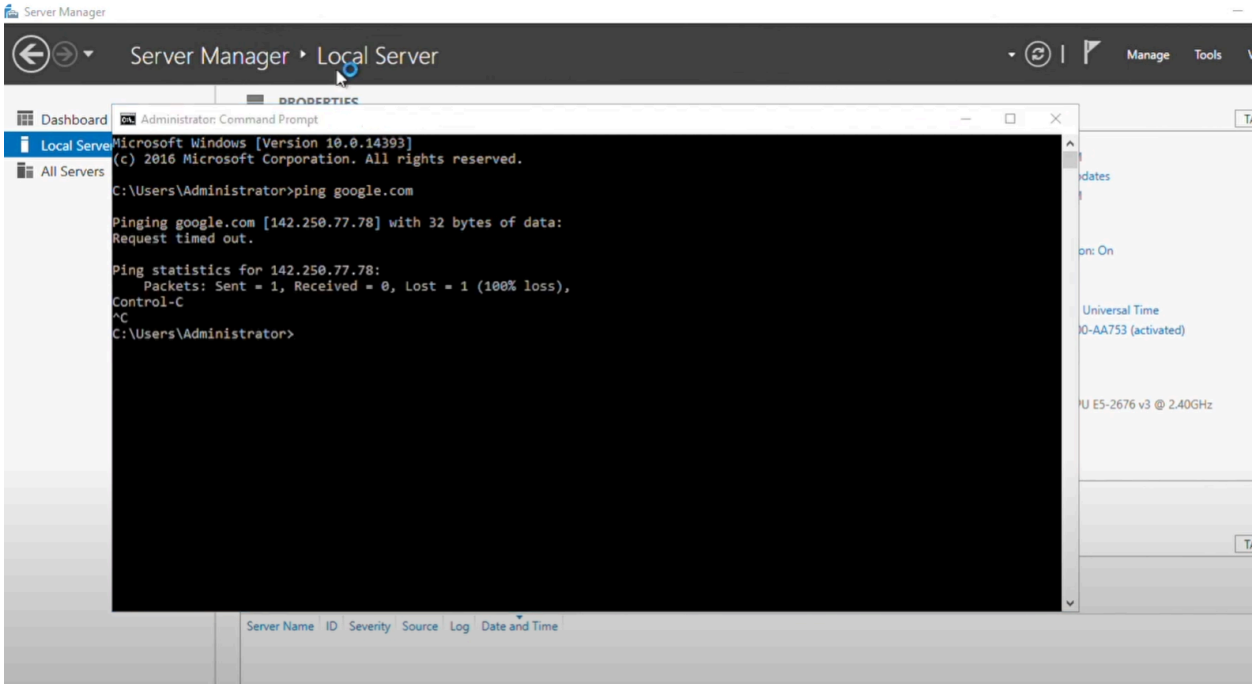


*Figure 4.4.1: Google Server Ping*
A FortiGate web filter successfully blocked access to online shopping, which is confirmed by a timed-out ping request to Google's servers.

From the Windows VM, a ping google.com command was executed. As expected, the ping requests timed out. This validated that the firewall policy correctly restricted outbound traffic to only HTTP and HTTPS, blocking ICMP (ping) traffic.

### 4.3 Web Browsing (YouTube)

Accessing youtube.com from the Windows VM's browser was successful. This confirmed general outbound internet connectivity for allowed services (HTTP/HTTPS) through the FortiGate.

### 4.4 Web Filtering Test (Amazon)

An attempt to browse amazon.in from the Windows VM resulted in a "FortiGuard Intrusion Prevention - Access Blocked" page. This conclusively demonstrated the effectiveness of the TESTWebFilter profile in blocking content categorized as "Shopping," as configured.

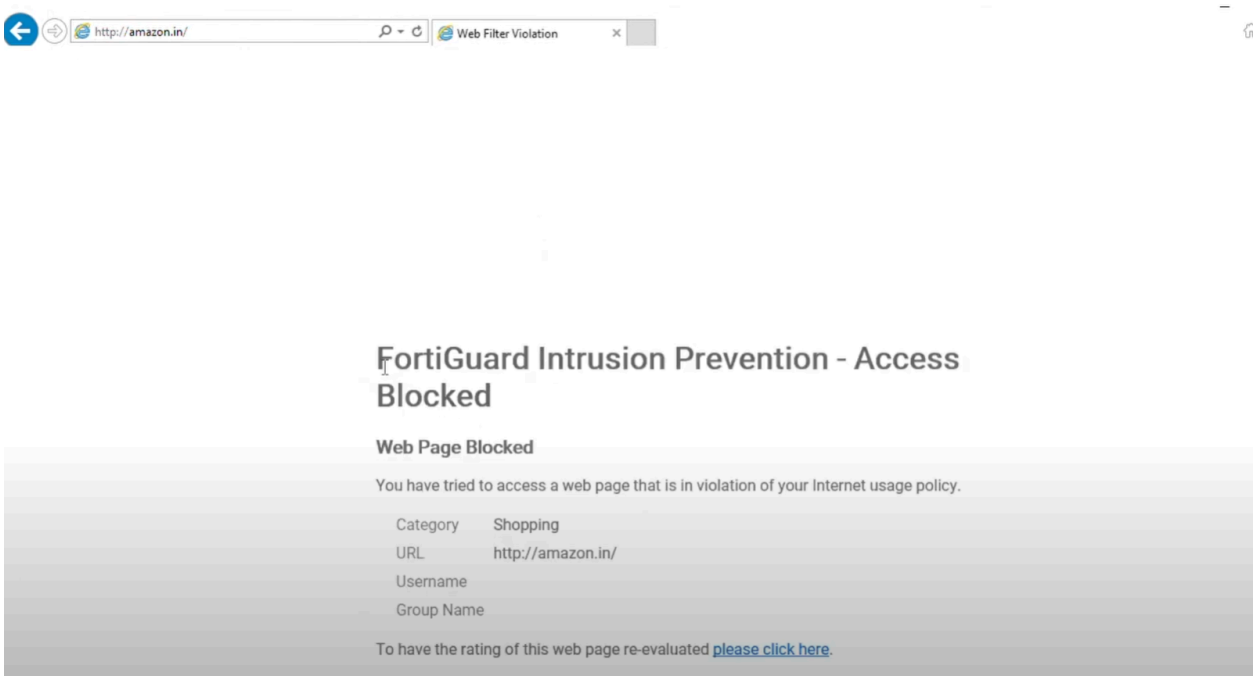


*Figure 4.4.1: Web Filter Block Page*
The figure shows a screenshot from a Windows VM's browser displaying a FortiGuard "Web Page Blocked" notification, indicating that the FortiGate's web filter successfully blocked access to amazon.in based on the "Shopping" category policy.

---

## 5. Conclusion

This project successfully demonstrated the deployment and configuration of a FortiGate firewall in an AWS environment to secure a Windows virtual machine. Key achievements include:

- Establishing a robust AWS network architecture with public and private subnets.
- Implementing the FortiGate as a central security gateway, routing all private subnet traffic through it.
- Configuring inbound RDP access to a private VM using FortiGate Virtual IPs.
- Enforcing granular outbound firewall policies, allowing specific services while blocking others.
- Successfully deploying and validating FortiGate's Unified Threat Management (UTM) capabilities, specifically web filtering, to block undesirable content categories.

This project showcases practical skills in cloud infrastructure deployment, network security, and firewall management, which are critical for securing modern cloud-based applications and data.