1.0.2 - ami-022ac2c9a86b21b5a

WireGuard VPN Server

A production-ready WireGuard VPN server with a web management interface for easy client configuration. Deploy a secure, high-performance VPN in minutes.

Overview

This AMI provides a complete WireGuard VPN solution using wg-easy, which includes:

- WireGuard VPN server
- Web-based management interface
- QR code generation for mobile clients
- Traffic statistics and monitoring
- Optional HTTPS with automatic Let's Encrypt certificates

Requirements

Resource	Minimum	Recommended
Instance Type	t3.micro	t3.small
vCPUs	1	2
RAM	512 MB	1 GB
EBS Storage	1 GB	5 GB

Ports

Port	Protocol	Purpose
22	TCP	SSH access
80	TCP	HTTP (for Let's Encrypt or redirect)
443	TCP	HTTPS (when enabled)

Port	Protocol	Purpose
51820	UDP	WireGuard VPN tunnel
51821	TCP	Web UI (HTTP mode only)

Important: Ensure your Security Group allows UDP port 51820 for VPN connectivity.

Quick Start

Option 1: Interactive Configuration

1. Launch the AMI with your desired instance type
2. Attach an EBS volume for persistent data (recommended)
3. Connect via SSH:

```
ssh -i your-key.pem ubuntu@your-instance-ip
```

4. Run the configuration script:

```
sudo /opt/wireguard/configure-wireguard.sh
```

5. Follow the prompts to set your admin password and configure options

Option 2: Automated Configuration (User-Data)

Launch the instance with the following JSON user-data:

```
{
  "admin_password": "your-secure-password-min-8-chars",
  "wg_host": "vpn.example.com",
  "use_https": true,
  "letsencrypt_email": "admin@example.com",
  "wg_dns": "1.1.1.1, 1.0.0.1",
  "wg_subnet": "10.8.0.x"
}
```

User-Data Parameters

Parameter	Required	Default	Description
admin_password	Yes	-	Web UI password (min 8 characters)

Parameter	Required	Default	Description
wg_host	No	Auto-detected IP	Domain or IP for VPN clients
use_https	No	false	Enable HTTPS with Let's Encrypt
letsencrypt_email	If HTTPS	-	Email for Let's Encrypt
wg_dns	No	1.1.1.1, 1.0.0.1	DNS servers for VPN clients
wg_subnet	No	10.8.0.x	VPN client IP range

Terraform Example

```
resource "aws_instance" "wireguard" {
  ami           = "ami-xxxxxxxxx"
  instance_type = "t3.micro"
  key_name      = "your-key"

  vpc_security_group_ids = [aws_security_group.wireguard.id]

  user_data = jsonencode({
    admin_password    = "your-secure-password"
    wg_host          = "vpn.example.com"
    use_https        = true
    letsencrypt_email = "admin@example.com"
  })

  tags = {
    Name = "WireGuard-VPN"
  }
}

resource "aws_security_group" "wireguard" {
  name        = "wireguard-sg"
  description = "WireGuard VPN Security Group"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["YOUR-IP/32"]
```

```
  }

  ingress {
    from_port   = 51820
    to_port     = 51820
    protocol    = "udp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

Management Commands

The `wireguard-cli` utility provides easy management:

```
# View service status
wireguard-cli status

# Start/stop/restart services
sudo wireguard-cli start
sudo wireguard-cli stop
sudo wireguard-cli restart

# View logs (use -f to follow)
wireguard-cli logs
```

```
wireguard-cli logs -f

# Show configuration info
wireguard-cli info

# List connected clients
wireguard-cli clients

# Update to latest version
sudo wireguard-cli update

# Create backup
sudo wireguard-cli backup

# Restore from backup
sudo wireguard-cli restore backup-file.tar.gz

# Reset admin password
sudo wireguard-cli reset-password
```

Creating VPN Clients

1. Access the web interface at your configured URL
2. Log in with your admin password
3. Click "New Client" and enter a name
4. Download the configuration file or scan the QR code

Client Installation

Desktop (Windows/macOS/Linux):

1. Download WireGuard from https://www.wireguard.com/install/
2. Import the configuration file

Mobile (iOS/Android):

1. Install WireGuard from App Store or Play Store
2. Scan the QR code displayed in the web interface

HTTPS Configuration

Using a Domain Name

For HTTPS with Let's Encrypt, you need:

1. A domain name pointing to your server's IP
2. Ports 80 and 443 accessible from the internet
3. A valid email address for certificate notifications

The configuration script will automatically set up nginx-proxy and acme-companion to handle certificate provisioning and renewal.

Using an IP Address

When using an IP address directly, HTTPS is not available. The web interface will be accessible on port 51821 over HTTP. Consider using a VPN client to access the admin interface securely.

Storage

EBS Volume (Recommended)

Attach an EBS volume before running configuration for persistent storage:

- VPN configurations survive instance replacement
- Client data persists across restarts
- Easy backup and migration

The configuration script automatically detects and formats EBS volumes at:

- /dev/xvdb
- /dev/xvdf
- /dev/nvme1n1
- /dev/sdf

Root Volume

Without an EBS volume, data is stored on the root volume. This is suitable for testing but not recommended for production.

Backup and Recovery

Creating a Backup

```
sudo wireguard-cli backup
```

This creates a timestamped archive in /tmp containing:

- WireGuard configuration and keys
- Client configurations
- Docker compose file

Restoring a Backup

```
sudo wireguard-cli restore /path/to/backup.tar.gz
```

Manual Backup

```
# Copy from the instance
scp -i your-key.pem ubuntu@server:/mnt/wireguard-data/config ./wireguard-backup/
```

Security Considerations

1. **Firewall:** Only open necessary ports in your Security Group
2. **SSH:** Restrict SSH access to known IP addresses
3. **Updates:** Regularly update the container images:

   ```
   sudo wireguard-cli update
   ```

4. **Passwords:** Use strong, unique passwords (minimum 12 characters recommended)
5. **Client Management:** Regularly review and remove unused clients

Troubleshooting

VPN Connection Issues

1. Verify Security Group allows UDP 51820:

```
# Test from client
nc -zvu your-server-ip 51820
```

2. Check WireGuard is running:

```
wireguard-cli status
```

3. View container logs:

```
wireguard-cli logs -f
```

Web Interface Not Loading

1. Check container status:

```
docker ps
```

2. Verify ports are listening:

```
sudo netstat -tlnp | grep -E '51821|443|80'
```

3. Check firewall rules:

```
sudo ufw status
```

SSL Certificate Issues

1. Ensure DNS is properly configured:

```
dig +short your-domain.com
```

2. Check acme-companion logs:

```
docker logs acme-companion
```

3. Certificate provisioning may take a few minutes. Wait and refresh.

Container Won't Start

1. Check for port conflicts:

```
sudo netstat -tlnp | grep 51820
```

2. Verify Docker is running:

```
sudo systemctl status docker
```

3. Review container logs:

```
docker logs wg-easy
```

File Locations

Path	Purpose
/opt/wireguard	Application directory
/opt/wireguard/docker-compose.yml	Container configuration
/opt/wireguard/config-info.txt	Setup information
/mnt/wireguard-data	Persistent data directory
/mnt/wireguard-data/config	WireGuard configurations
/mnt/wireguard-data/certs	SSL certificates
/var/log/wireguard-firstboot.log	First boot log

Support

For issues specific to this AMI, please contact Propagate support.

For WireGuard and wg-easy documentation:

- WireGuard: https://www.wireguard.com/
- wg-easy: https://github.com/wg-easy/wg-easy

Revision #2
Created 2026-01-23 17:23:18 UTC by Admin
Updated 2026-01-23 17:34:44 UTC by Admin