

Hướng Dẫn: AWS Site-to-Site VPN - Kết nối On-Premise với Amazon VPC

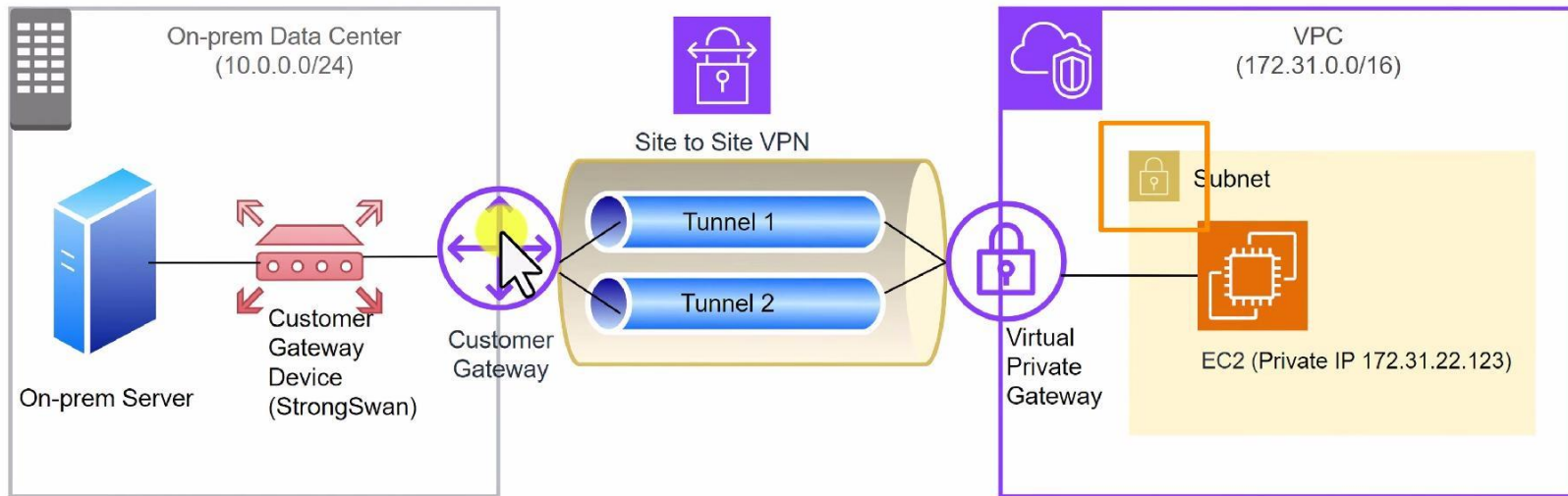
BƯỚC 1

Mô hình: Bên trái là On-premise Data Center (gồm Raspberry Pi làm Customer Gateway cài StrongSwan VPN, và 1 server). Bên phải là AWS VPC với EC2 instance. Mục tiêu: thiết lập Site-to-Site VPN kết nối 2 bên qua IPSec tunnel.



BƯỚC 2

Trên AWS cần tạo: Customer Gateway (đại diện thiết bị on-prem), Virtual Private Gateway (điểm vào/ra VPN traffic cho VPC), và VPN Connection. Cấu hình 2 tunnel (Tunnel 1 & Tunnel 2) để đảm bảo redundancy và high availability.



BƯỚC 3

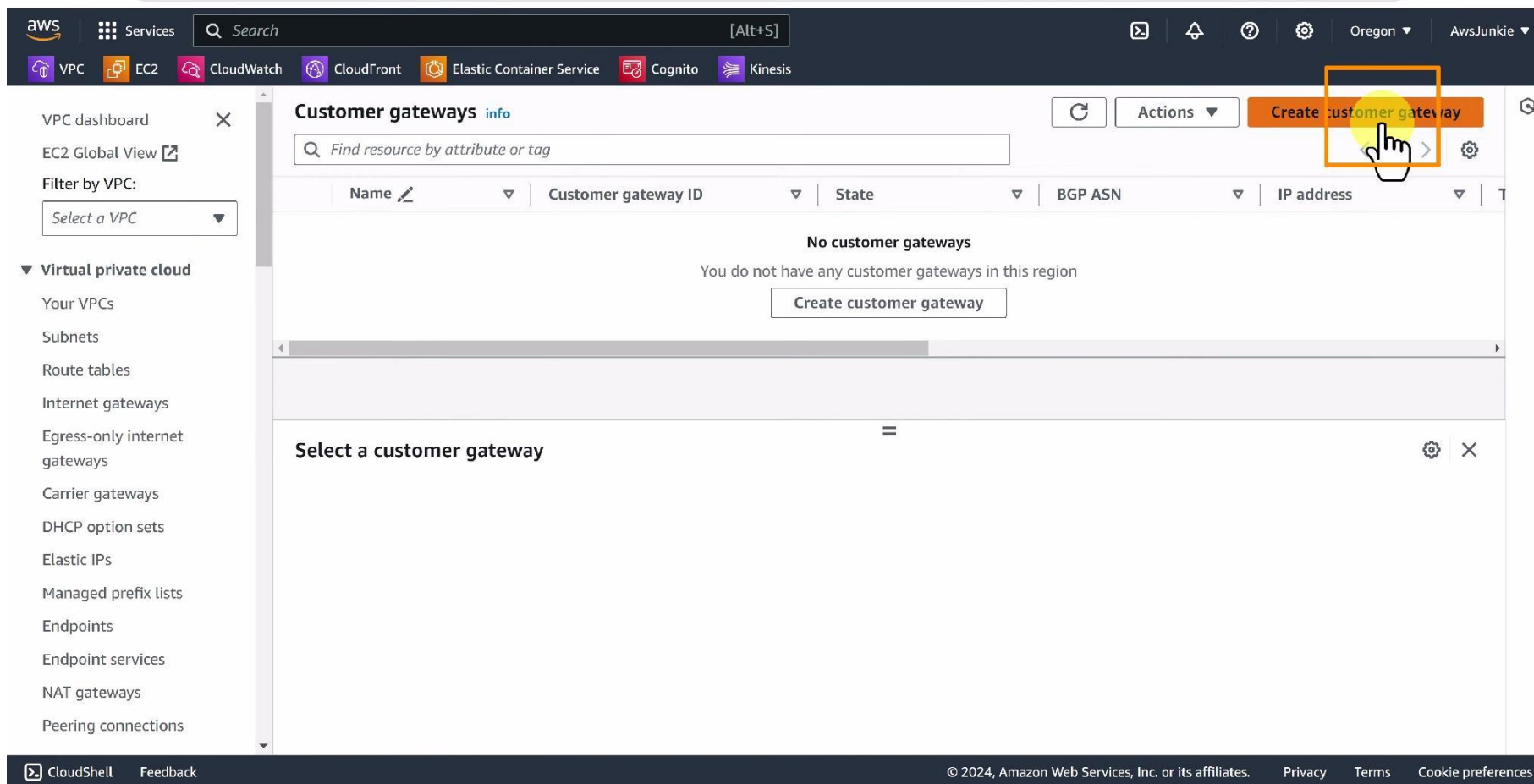
Đăng nhập AWS Console, mở VPC Dashboard. Trong menu bên trái phần Virtual Private Network, tìm các mục: Customer Gateways, Virtual Private Gateways, Site-to-Site VPN Connections.

The screenshot displays the AWS Management Console VPC dashboard for the us-west-2 region. The top navigation bar includes the AWS logo, a search bar, and a list of services. The left sidebar shows the VPC dashboard and various navigation links. The main content area is titled 'Resources by Region' and displays a grid of VPC resources. The right sidebar contains sections for Service Health, Settings, Additional Information, and AWS Network Manager. A pink box highlights the address bar showing 'us-west-2.console.aws.amazon.com'.

Resource	US West
VPCs	1
NAT Gateways	0
Subnets	4
VPC Peering Connections	0
Route Tables	1
Network ACLs	1
Internet Gateways	1
Security Groups	1
Egress-only Internet Gateways	0
Customer Gateways	0

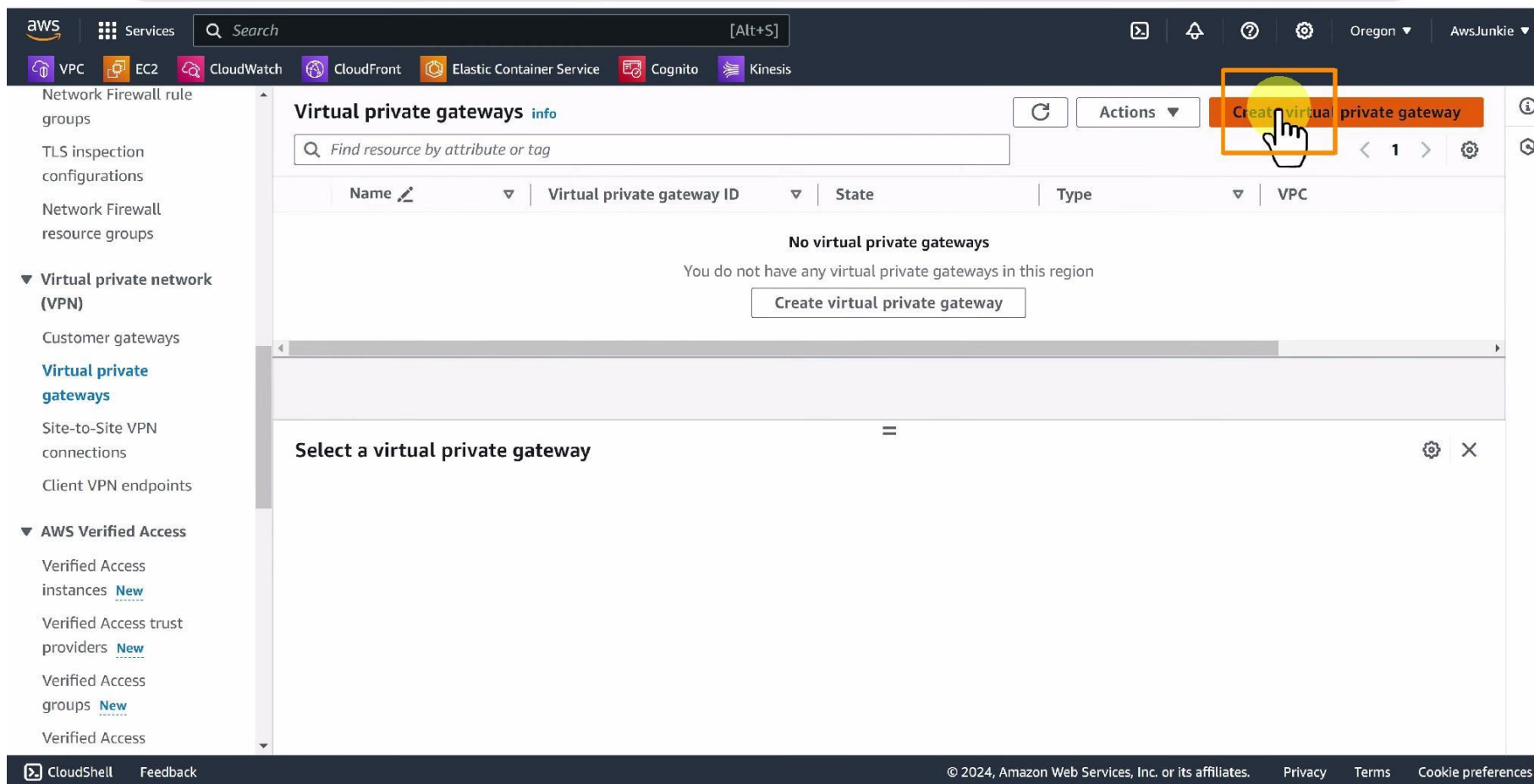
BƯỚC 4

Tạo Customer Gateway: Click 'Customer Gateways' → 'Create Customer Gateway'. Đặt tên (ví dụ: CGW-1). Nhập Public IP address của thiết bị Customer Gateway (IP public của router/firewall on-prem). Click Create.



BƯỚC 5

Tạo Virtual Private Gateway: Click 'Virtual Private Gateways' → 'Create Virtual Private Gateway'. Đặt tên (VPG-1). Để ASN mặc định (Amazon default ASN). Click Create.



BƯỚC 6

Attach Virtual Private Gateway vào VPC: Chọn VPG vừa tạo (đang ở trạng thái Detached) → Actions → Attach to VPC → Chọn VPC cần kết nối → Click Attach. Đợi trạng thái chuyển sang Attached.

aws Services Search [Alt+S] Oregon AwsJunkie

VPC EC2 CloudWatch CloudFront Elastic Container Service Cognito Kinesis

VPC dashboard X

EC2 Global View

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

You successfully created vgw-02d840c9a50546c5e / vpg-1.

Virtual private gateways (1/1) info

Find resource by attribute or tag

Virtual private gateway ID = vgw-02d840c9a50546c5e X Clear filters

Actions Create virtual private gateway

Name	Virtual private gateway ID	State	Type	VPC
vpg-1	vgw-02d840c9a50546c5e	Detached	ipsec.1	-

Virtual private gateway vgw-02d840c9a50546c5e / vpg-1

Details Tags

Details

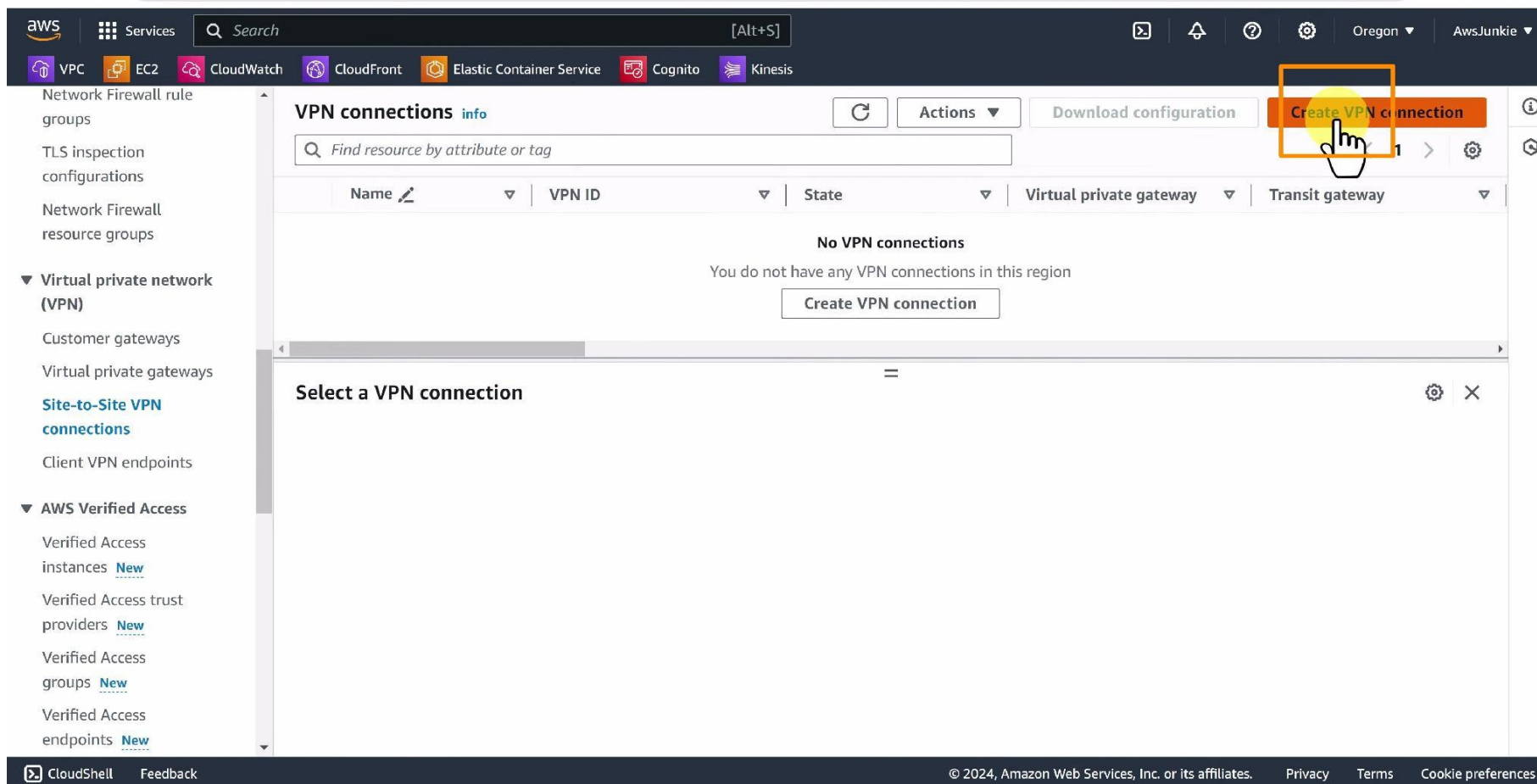
Virtual private gateway ID	State	Type	VPC
vgw-02d840c9a50546c5e	Detached	ipsec.1	-
Amazon ASN			
64512			

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

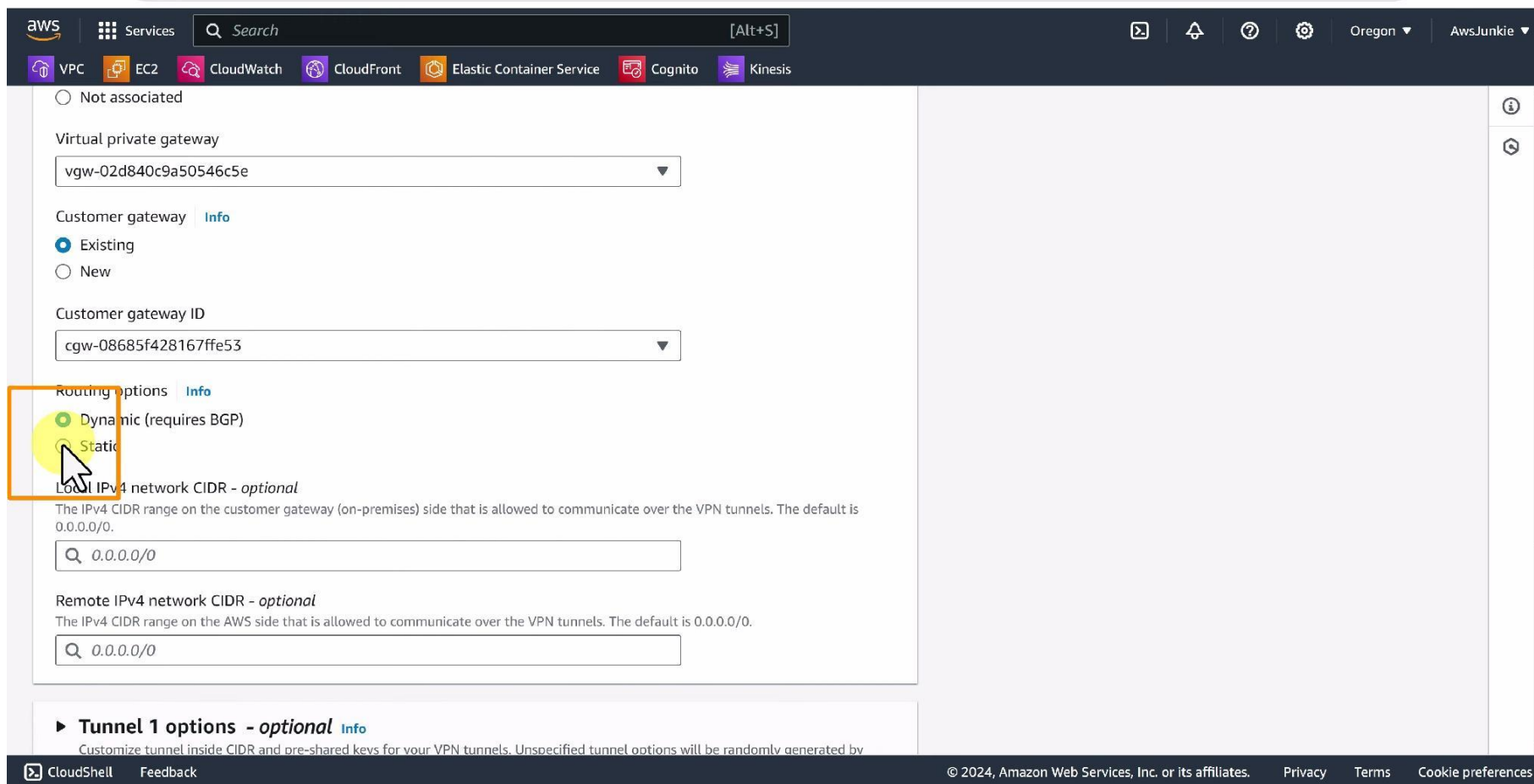
BƯỚC 7

Tạo VPN Connection: Click 'Site-to-Site VPN Connections' → 'Create VPN Connection'. Đặt tên (VPN-1). Target Gateway: chọn Virtual Private Gateway vừa tạo. Customer Gateway: chọn Existing → chọn CGW đã tạo.



BƯỚC 8

Cấu hình Routing: Vì không có BGP, chọn 'Static'. Nhập Static IP Prefix = CIDR range của mạng on-prem (ví dụ: 10.1.0.0/24). Click 'Create VPN Connection'.



BƯỚC 9

Kiểm tra VPN Connection: Click vào tab Tunnel Details. Thấy 2 tunnel đã được tạo nhưng trạng thái đang Down (vì chưa cấu hình phía on-prem).

aws Services Search [Alt+S] Oregon AwsJunkie

VPC EC2 CloudWatch CloudFront Elastic Container Service Cognito Kinesis

VPC dashboard X

EC2 Global View

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

You successfully created vpn-07c1b79c0dfdb3da5 / vpn-1.

VPN connections (1/1) info

Find resource by attribute or tag

VPN ID = vpn-07c1b79c0dfdb3da5 X Clear filters

< 1 >

Name	VPN ID	State	Virtual private gateway	Transit gateway
vpn-1	vpn-07c1b79c0dfdb3da5	Pending	vgw-02d840c9a50546c5e	-

VPN connection vpn-07c1b79c0dfdb3da5 / vpn-1

Details Tunnel details Static routes Tags

Tunnel state

Tunnel number	Outside IP address	Inside IPv4 CIDR	Inside IPv6 CIDR	Status	Last status change	Detail
Tunnel 1	44.237.39.77	169.254.141.112/30	-	Down	April 15, 2024, 20:40:34 (UTC-04:00)	-
Tunnel 2	52.26.100.233	169.254.127.188/30	-	Down	April 15, 2024, 20:40:34 (UTC-04:00)	-

Tunnel 1 options Info

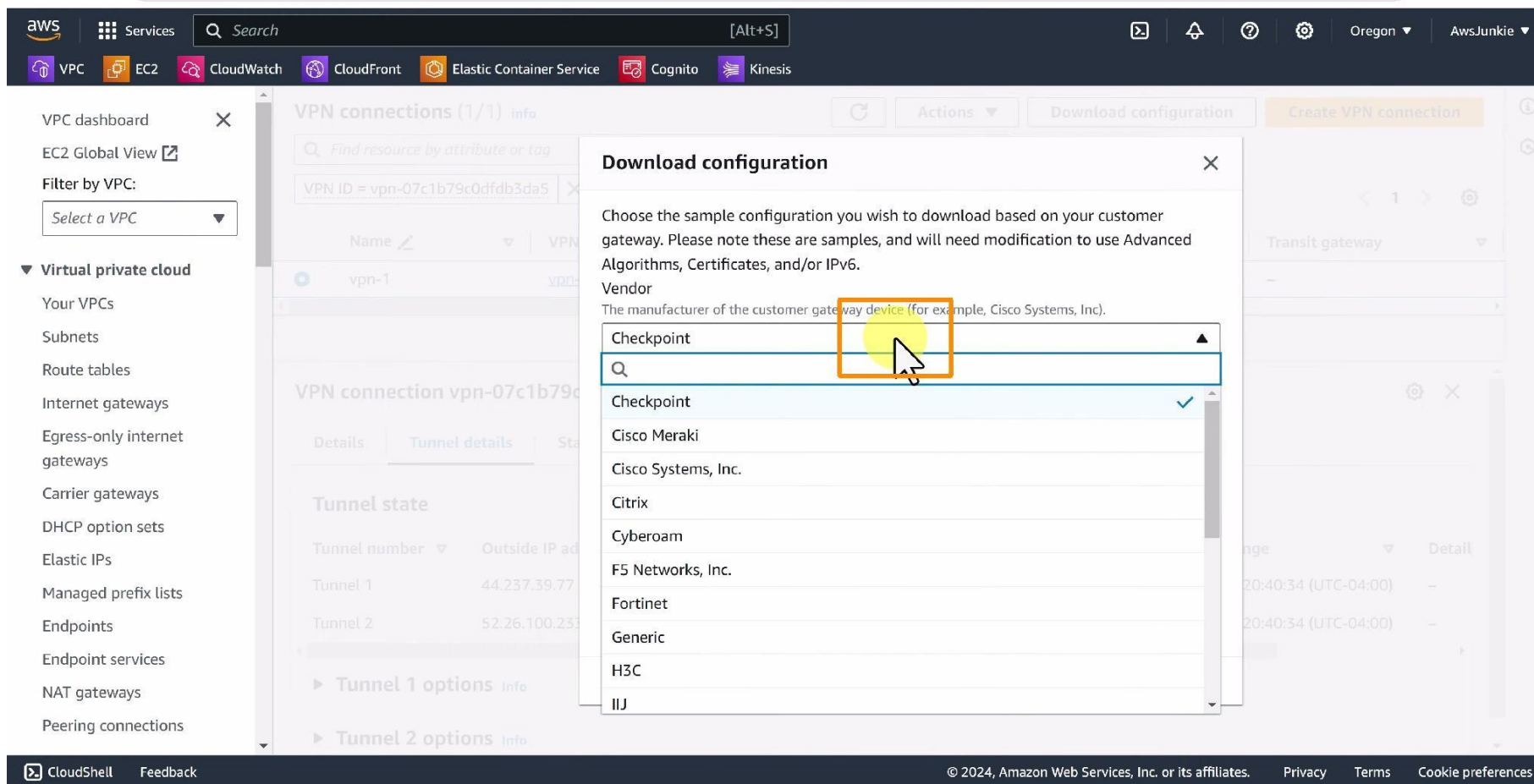
Tunnel 2 options Info

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

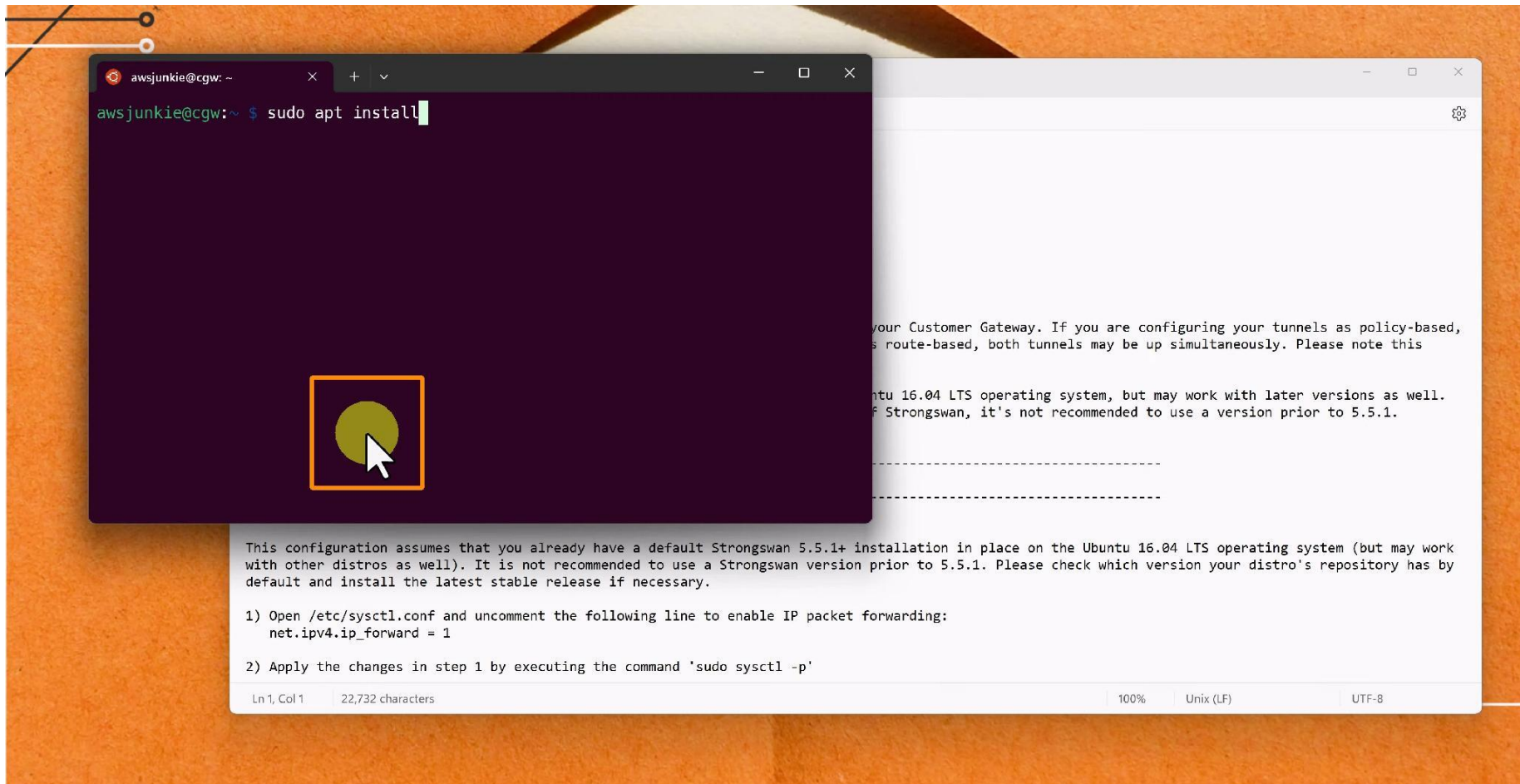
BƯỚC 10

Download Configuration: Click 'Download Configuration'. Chọn Vendor = StrongSwan (hoặc vendor phù hợp với thiết bị của bạn). File này chứa toàn bộ thông tin cần cấu hình trên Customer Gateway device.



BƯỚC 11

SSH vào Customer Gateway device (Raspberry Pi). Cài đặt StrongSwan: chạy lệnh 'sudo apt install strongswan'.



BƯỚC 12

Bật IP Forwarding: Mở file /etc/sysctl.conf, uncomment dòng 'net.ipv4.ip_forward=1'. Lưu file. Apply bằng lệnh 'sudo sysctl -p'.

```
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

#####
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1
```

[Read 68 lines]

^G Help

^X Exit

^O Write Out

^R Read File

^W Where Is

^_ Replace

^K Cut

^U Paste

^T Execute

^J Justify

```
-----
IPSEC Tunnel #1
-----

#1: Enable Packet Forwarding and Configure the Tunnel

This configuration assumes that you already have a default Strongswan 5.5.1+ installation
in place on the Ubuntu 16.04 LTS operating system (but may work with other distros as
well). It is not recommended to use a Strongswan version prior to 5.5.1. Please check which
version your distro's repository has by default and install the latest stable release if
necessary.

1) Open /etc/sysctl.conf and uncomment the following line to enable IP packet forwarding:
   net.ipv4.ip_forward = 1

2) Apply the changes in step 1 by executing the command 'sudo sysctl -p'

3) Create a new file at /etc/ipsec.conf if doesn't already exist, and then open it.
Uncomment the line "uniqueids=no" under the 'config setup' section. Append the following
configuration to the end of the file:

# AWS VPN will also support AES256 and SHA256 for the "ike" (Phase 1) and "esp" (Phase 2)
entries below.
# For Phase 1, AWS VPN supports DH groups 2, 14-18, 22, 23, 24. Phase 2 supports DH groups
2, 5, 14-18, 22, 23, 24
# To see Strongswan's syntax for these different values, please refer to
https://wiki.strongswan.org/projects/strongswan/wiki/IKEv1CipherSuites

conn Tunnel1
    auto=start
    left=%defaultroute
    leftid=137.194.247.33
    right=44.237.39.77
    type=tunnel
    leftauth=psk
    rightauth=psk
    keyexchange=ikev1
    ike=aes128-sha1-modp1024
    ikelifetime=8h
    esp=aes128-sha1-modp1024
    lifetime=1h
    keyingtries=%forever
```

Ln 26, Col 25 | 16 of 22,732 characters | 100% | Unix (LF) | UTF-8

BƯỚC 13

Cấu hình IPsec: Mở file /etc/ipsec.conf (tạo mới nếu chưa có). Uncomment dòng 'uniqueids = no'.

```
awsjunkie@cgw:~ $ sudo sysctl -p
net.ipv4.ip_forward = 1
awsjunkie@cgw:~ $
```

IPSEC Tunnel #1

#1: Enable Packet Forwarding and Configure the Tunnel

This configuration assumes that you already have a default Strongswan 5.5.1+ installation in place on the Ubuntu 16.04 LTS operating system (but may work with other distros as well). It is not recommended to use a Strongswan version prior to 5.5.1. Please check which version your distro's repository has by default and install the latest stable release if necessary.

1) Open /etc/sysctl.conf and uncomment the following line to enable IP packet forwarding:
net.ipv4.ip_forward = 1

2) Apply the changes in step 1 by executing the command 'sudo sysctl -p'

3) Create a new file at /etc/ipsec.conf if it doesn't already exist, and then open it. Uncomment the line "uniqueids=no" under the 'config setup' section. Append the following configuration to the end of the file:

```
# AWS VPN will also support AES256 and SHA256 for the "ike" (Phase 1) and "esp" (Phase 2)
entries below.
# For Phase 1, AWS VPN supports DH groups 2, 14-18, 22, 23, 24. Phase 2 supports DH groups
2, 5, 14-18, 22, 23, 24
# To see Strongswan's syntax for these different values, please refer to
https://wiki.strongswan.org/projects/strongswan/wiki/IKEv1CipherSuites
```

```
conn Tunnel1
    auto=start
    left=%defaultroute
    leftid=137.194.247.33
    right=44.237.39.77
    type=tunnel
    leftauth=psk
    rightauth=psk
    keyexchange=ikev1
    ike=aes128-sha1-modp1024
    ikelifetime=8h
    esp=aes128-sha1-modp1024
    lifetime=1h
    keyingtries=%forever
```

Ln 31, Col 25 24 of 22,732 characters

100%

Unix (LF)

UTF-8

BƯỚC 14

Copy cấu hình Tunnel 1 từ file configuration đã tải: Uncomment dòng 'leftupdown', thay CIDR range VPC side vào đúng vị trí. Copy toàn bộ section 'conn Tunnel1' và paste vào cuối file ipsec.conf.

```
# ipsec.conf - strongSwan IPsec configuration file
```

```
# basic configuration
```

```
config setup
```

```
# strictcrpolicies=yes
```

```
uniqueids = no
```

```
# Add connections here.
```

```
# Sample VPN connections
```

```
#conn sample-self-signed
```

```
# leftsubnet=10.1.0.0/16
```

```
# leftcert=selfCert.der
```

```
# leftsendcert=never
```

```
# right=192.168.0.2
```

```
# rightsubnet=10.2.0.0/16
```

```
# rightcert=peerCert.der
```

```
# auto=start
```

```
#conn sample-with-ca-cert
```

```
# leftsubnet=10.1.0.0/16
```

```
# leftcert=myCert.pem
```

```
# right=192.168.0.2
```

```
# rightsubnet=10.2.0.0/16
```

```
# rightid="C=CH, O=Linux strongSwan CN=peer name"
```

```
# auto=start
```

```
[ Read 29 lines ]
```

```
^G Help
```

```
^O Write Out
```

```
^W Where Is
```

```
^K Cut
```

```
^T Execute
```

```
^X Exit
```

```
^R Read File
```

```
^\ Replace
```

```
^U Paste
```

```
^J Justify
```

necessary.

1) Open /etc/sysctl.conf and uncomment the following line to enable IP packet forwarding:
net.ipv4.ip_forward = 1

2) Apply the changes in step 1 by executing the command 'sudo sysctl -p'

3) Create a new file at /etc/ipsec.conf if doesn't already exist, and then open it. Uncomment the line "uniqueids=no" under the 'config setup' section. Append the following configuration to the end of the file:

```
# AWS VPN will also support AES256 and SHA256 for the "ike" (Phase 1) and "esp" (Phase 2) entries below.
```

```
# For Phase 1, AWS VPN supports DH groups 2, 14-18, 22, 23, 24. Phase 2 supports DH groups 2, 5, 14-18, 22, 23, 24
```

```
# To see Strongswan's syntax for these different values, please refer to  
https://wiki.strongswan.org/projects/strongswan/wiki/IKEv1CipherSuites
```

```
conn Tunnel1
```

```
auto=start
```

```
left=%defaultroute
```

```
leftid=137.194.247.33
```

```
right=44.237.39.77
```

```
type=tunnel
```

```
leftauth=psk
```

```
rightauth=psk
```

```
keyexchange=ikev1
```

```
ike=aes128-sha1-modp1024
```

```
ikelifetime=8h
```

```
esp=aes128-sha1-modp1024
```

```
lifetime=1h
```

```
keyingtries=%forever
```

```
leftsubnet=0.0.0.0/0
```

```
rightsubnet=0.0.0.0/0
```

```
updelay=10s
```

```
uplifetime=30s
```

```
upaction=restart
```

```
## Please note the following line assumes you only have two tunnels in your  
Strongswan configuration file. This "mark" value must be unique and may need to be changed  
based on other entries in your configuration file.
```

```
mark=100
```

```
## Uncomment the following line to utilize the script from the "Automated Tunnel  
Healthcheck and Failover" section. Ensure that the integer after "-m" matches the "mark"  
value above, and <VPC CIDR> is replaced with the CIDR of your VPC
```

Ln 52, Col 2 640 of 22,732 characters

100%

Unix (LF)

UTF-8

BƯỚC 15

Copy cấu hình Tunnel 2: Làm tương tự - uncomment 'leftupdown', thay CIDR range, copy toàn bộ section 'conn Tunnel2' và paste vào cuối file ipsec.conf.

Lưu file.

```

ikelifetime=8h
esp=aes128-sha1-modp1024
lifetime=1h
keyingtries=%forever
leftsubnet=0.0.0.0/0
rightsubnet=0.0.0.0/0
dpddelay=10s
dpdtimeout=30s
dpdaction=restart
## Please note the following line assumes you only have two tunnels>
mark=100
## Uncomment the following line to utilize the script from the "Aut>
## (e.g. 192.168.1.0/24)
leftupdown="/etc/ipsec.d/aws-updown.sh -ln Tunnel1 -ll 169.254.141.>

```

```

keyexchange=ikev1
ike=aes128-sha1-modp1024
ikeexchange=ikev1
ike=aes128-sha1-modp1024
ikelifetime=8h
esp=aes128-sha1-modp1024
lifetime=1h
keyingtries=%forever
leftsubnet=0.0.0.0/0
rightsubnet=0.0.0.0/0
dpddelay=10s
dpdtimeout=30s
dpdaction=restart
## Please note the following line assumes you only have two tunnels in your
Strongswan configuration file. This "mark" value must be unique and may need to be changed
based on other entries in your configuration file.
mark=200
## Uncomment the following line to utilize the script from the "Automated Tunnel
Healthcheck and Failover" section. Ensure that the integer after "-m" matches the "mark"
value above, and <VPC CIDR> is replaced with the CIDR of your VPC
## (e.g. 192.168.1.0/24)
#leftupdown="/etc/ipsec.d/aws-updown.sh -ln Tunnel2 -ll 169.254.127.190/30 -lr
169.254.127.189/30 -m 200 -r <VPC CIDR>"

```

4) Create a new file at /etc/ipsec.secrets if it doesn't already exist, and append this line to the file (be mindful of the spacing!). This value authenticates the tunnel endpoints:

```
137.194.247.33 52.26.100.233 : PSK "70LSv0VtHbMhCsmH7T1e_h12RxaCwFGt_"
```

5) If you would like to configure your route-based tunnels manually, please complete the following steps #2 - #5. These steps may be omitted if you decide to follow the steps in the "Automated Tunnel Healthcheck and Failover" section of the document.

#2: Tunnel Interface Configuration

A tunnel interface is a logical interface associated with tunnel traffic. All traffic to/from the VPC will be logically transmitted and received by the tunnel interface.

1) If your device is in a VPC or behind a device performing NAT on your local network, replace <LOCAL IP> with the private IP of the device. Otherwise, use 137.194.247.33. The "key" value below MUST match the integer you placed as the "mark" value in your configuration file.

```
sudo ip link add Tunnel2 type vti local <LOCAL IP> remote 52.26.100.233 key 200
```

Ln 60, Col 122 13 of 22,734 characters

100%

Unix (LF)

UTF-8

[^]G Help [^]O Write Out [^]W Where Is [^]K Cut [^]T Execute
[^]X Exit [^]R Read File [^]\ Replace [^]U Paste [^]J Justify

BƯỚC 16

Cấu hình Shared Secrets: Mở file /etc/ipsec.secrets (tạo mới nếu chưa có). Copy Pre-Shared Key của Tunnel 1 và Tunnel 2 từ file configuration, paste vào file ipsec.secrets. Lưu file.

```
awsjunkie@cgw:~ $ sudo sysctl -p
net.ipv4.ip_forward = 1
awsjunkie@cgw:~ $ sudo nano /etc/ipsec.conf
awsjunkie@cgw:~ $
```



```
right=44.237.39.77
type=tunnel
leftauth=psk
rightauth=psk
keyexchange=ikev1
ike=aes128-sha1-modp1024
ikelifetime=8h
esp=aes128-sha1-modp1024
lifetime=1h
keyingtries=%forever
leftsubnet=0.0.0.0/0
rightsubnet=0.0.0.0/0
dpddelay=10s
dpdtimeout=30s
dpdaction=restart
## Please note the following line assumes you only have two tunnels in your
Strongswan configuration file. This "mark" value must be unique and may need to be changed
based on other entries in your configuration file.
mark=100
## Uncomment the following line to utilize the script from the "Automated Tunnel
Healthcheck and Failover" section. Ensure that the integer after "-m" matches the "mark"
value above, and <VPC CIDR> is replaced with the CIDR of your VPC
## (e.g. 192.168.1.0/24)
leftupdown="/etc/ipsec.d/aws-updown.sh -ln Tunnel1 -l1 169.254.141.114/30 -lr
169.254.141.113/30 -m 100 -r 172.31.0.0/16"

4) Create a new file at /etc/ipsec.secrets if it doesn't already exist, and append this
line to the file (be mindful of the spacing!). This value authenticates the tunnel
endpoints:
137.194.247.33 44.237.39.77 : PSK "3jvboSF6utAOel3xjoCMqgKXUrwC_8rL"

5) If you would like to configure your route-based tunnels manually, please complete the
following steps #2 - #5. These steps may be omitted if you decide to follow the steps in
the "Automated Tunnel Healthcheck and Failover" section of the document.
```

----- #2: Tunnel Interface Configuration

A tunnel interface is a logical interface associated with tunnel traffic. All traffic to/from the VPC will be logically transmitted and received by the tunnel interface.

1) If your device is in a VPC or behind a device performing NAT on your local network, replace <LOCAL IP> with the private IP of the device. Otherwise, use 137.194.247.33. The "key" value below MUST match the integer you placed as the "mark" value in your

Ln 62, Col 43 | 18 of 22,736 characters

100%

Unix (LF)

UTF-8

BƯỚC 17

Tạo script Up/Down: Tạo file /etc/ipsec.d/aws-updown.sh. Copy toàn bộ script từ section 'Automated Tunnel Healthcheck and Failover' trong file config. Thêm dòng 'src [local IP]' vào cuối lệnh 'ip route add'. Lưu file và chmod +x để cấp quyền thực thi.

```
awsjunkie@cgw:~ $ sudo systemctl -p
net.ipv4.ip_forward = 1
awsjunkie@cgw:~ $ sudo nano /etc/ipsec.conf
awsjunkie@cgw:~ $ sudo nano /etc/ipsec.secrets
awsjunkie@cgw:~ $
```

the differences, please refer to
<http://packetlife.net/blog/2011/aug/15/policy-based-vs-route-based-vpns-part-1/>

=== DISCLAIMER ===

Please be aware that AWS is in no way responsible for any of the use, management, maintenance, or potential issues you may encounter with the following tunnel failover workaround. It is strongly recommended that you thoroughly test any failover solution prior to implementing it into your production environment.

This failover mechanism has been tested using Strongswan 5.5.1 and Ubuntu 16.04 LTS.

=== HOW-TO ===

1) Create a new file at `/etc/ipsec.d/aws-updown.sh` if it doesn't already exist, and append the following script to the file:

```
#!/bin/bash

while [[ $# > 1 ]]; do
  case ${1} in
    -ln|--link-name)
      TUNNEL_NAME="${2}"
      TUNNEL_PHY_INTERFACE="${PLUTO_INTERFACE}"
      shift
      ;;
    -ll|--link-local)
      TUNNEL_LOCAL_ADDRESS="${2}"
      TUNNEL_LOCAL_ENDPOINT="${PLUTO_ME}"
      shift
      ;;
    -lr|--link-remote)
      TUNNEL_REMOTE_ADDRESS="${2}"
      TUNNEL_REMOTE_ENDPOINT="${PLUTO_PEER}"
      shift
      ;;
    -m|--mark)
      TUNNEL_MARK="${2}"
      shift
      ;;
    -r|--static-route)
      TUNNEL_STATIC_ROUTE="${2}"
      shift
      ;;
    *)
      echo "${0}: Unknown argument \"${1}\"" >&2
  esac
  shift
done
```

Ln 275, Col 1 22,736 characters

100%

Unix (LF)

UTF-8

BƯỚC 18

Khởi động lại StrongSwan: Chạy 'sudo ipsec restart'. Kiểm tra trạng thái: 'sudo ipsec status'. Kết quả hiển thị Tunnel 1 và Tunnel 2 đều ESTABLISHED.

```
awsjunkie@cgw:~ $ sudo sysctl -p
net.ipv4.ip_forward = 1
awsjunkie@cgw:~ $ sudo nano /etc/ipsec.conf
awsjunkie@cgw:~ $ sudo nano /etc/ipsec.secrets
awsjunkie@cgw:~ $ sudo nano /etc/ipsec.d/aws-updown.sh
awsjunkie@cgw:~ $ sudo chmod 744 /etc/ipsec.d/aws-updown.sh
awsjunkie@cgw:~ $
```

```
create_interface
configure_sysctl
add_route
;;
down-client)
cleanup
delete_interface
;;
esac
```

- 2) To make the file executable, run the command 'sudo chmod 744 /etc/ipsec.d/aws-updown.sh'
- 3) Open the file /etc/ipsec.conf and ensure the "leftupdown" parameter at the end of each of your 'conn' entries is uncommented. You will need to modify <VPC CIDR> to match the CIDR of your VPC (e.g. 192.168.1.0/24). Please also verify the integer value after the "-m" option matches the 'mark' parameter of your configuration if you have made changes to the default values of this configuration file.
- 4) Restart the Strongswan daemon by executing the command 'sudo ipsec restart'
- 5) Check if your updown script worked properly. You can use the following commands to test if there are entries created for each of your tunnels:
 - Execute 'sudo ipsec status' to ensure both of your tunnels are ESTABLISHED
 - Execute 'sudo ip route' to ensure route table entries were created for each of your tunnel interfaces, and the destination is the remote VPC CIDR
 - Execute 'sudo iptables -t mangle -L -n' to ensure entries were made for both of your tunnels in both the INPUT and FORWARD chains
 - Execute 'ifconfig' to ensure the correct 169.254.x addresses were assigned to each end of your peer-to-peer virtual tunnel interfaces
 - Attempt to ping a destination in the remote VPC from a host within your local network. If there is no response, check to see if your instance's security groups are allowing traffic and verify your settings entered above are correct once again
- 6) Verify failover is working properly. You can test this by blocking traffic from the remote virtual private gateway (VGW) public IPs. For example:


```
sudo iptables -A INPUT -s <VGW PUBLIC IP> -j DROP
```

Additional Notes and Questions

- Amazon Virtual Private Cloud Getting Started Guide:
<http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide>
- Amazon Virtual Private Cloud Network Administrator Guide:
<http://docs.amazonwebservices.com/AmazonVPC/latest/NetworkAdminGuide>

Ln 371, Col 91 | 41 of 22,736 characters

100%

Unix (LF)

UTF-8

BƯỚC 19

Xác nhận trên AWS Console: Quay lại VPN Connection → Click Refresh → Chọn VPN → Tab Tunnel Details: cả 2 tunnel đều hiện trạng thái UP.

The screenshot shows the AWS Management Console interface for VPN connections. The left sidebar contains navigation links for various AWS services, including VPC, EC2, CloudWatch, CloudFront, Elastic Container Service, Cognito, and Kinesis. The main content area displays the 'VPN connections (1/1)' page. A table lists the VPN connections, with 'vpn-1' selected. Below the table, the 'VPN connection vpn-07c1b79c0dfdb3da5 / vpn-1' details are shown. The 'Tunnel details' tab is active, displaying a table of tunnel states. Two tunnels are listed, both with a 'Down' status, which is highlighted by a red box and a mouse cursor.

Name	VPN ID	State	Virtual private gateway	Transit gateway
vpn-1	vpn-07c1b79c0dfdb3da5	Available	vgw-02d840c9a50546c5e	-

Tunnel number	Outside IP address	Inside IPv4 CIDR	Inside IPv6 CIDR	Status	Last status change	Detail
Tunnel 1	44.237.39.77	169.254.141.112/30	-	Down	April 15, 2024, 21:08:58 (UTC-04:00)	-
Tunnel 2	52.26.100.233	169.254.127.188/30	-	Down	April 15, 2024, 21:09:00 (UTC-04:00)	-

BƯỚC 20

Tạo EC2 Instance để test: Launch Instance, đặt tên, proceed without key pair. Edit Network Settings: Disable auto-assign public IP. Security Group: Allow All ICMP từ CIDR range của on-prem data center. Click Launch.

aws Services [Alt+S]

VPC EC2 CloudWatch CloudFront Elastic Container Service Cognito Kinesis

EC2 > Instances > Launch an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name [Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

▼ Summary

Number of instances [Info](#)

Software Image (AMI)
Amazon Linux 2023 AMI 2023.4.2...[read more](#)
ami-0395649fbe870727e

Virtual server type (instance type)
-

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

[Free tier](#): In your first year includes 750 hours of t2.micro (or)

[Cancel](#) [Launch instance](#) [Review commands](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

BƯỚC 21

Cấu hình Route Table trong VPC: Vào VPC → Route Tables → Chọn route table → Tab Routes → Edit Routes → Add Route. Destination = CIDR on-prem (10.1.0.0/24), Target = Virtual Private Gateway. Save changes.

VPN connections (1/1) [info](#)

Find resource by attribute or tag

Name	VPN ID	State	Virtual private gateway	Transit gateway
vpn-1	vpn-07c1b79c0dfdb3da5	Available	vgw-02d840c9a50546c5e	-

VPN connection vpn-07c1b79c0dfdb3da5 / vpn-1

Details | **Tunnel details** | Static routes | Tags

Tunnel state

Tunnel number	Outside IP address	Inside IPv4 CIDR	Inside IPv6 CIDR	Status	Last status change	Detail
Tunnel 1	44.237.39.77	169.254.141.112/30	-	Up	April 15, 2024, 21:11:47 (UTC-04:00)	-
Tunnel 2	52.26.100.233	169.254.127.188/30	-	Up	April 15, 2024, 21:12:16 (UTC-04:00)	-

► **Tunnel 1 options** [info](#)

► **Tunnel 2 options** [info](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

BƯỚC 22

[KIỂM TRA] Copy Private IP của EC2 instance. Từ terminal trên Customer Gateway (on-prem), chạy lệnh 'ping [Private IP EC2]'. Kết quả: Ping thành công qua private IP → VPN Site-to-Site hoạt động!

aws

Services

Search

[Alt+S]

VPC

EC2

CloudWatch

CloudFront

Elastic Container Service

Cognito

Kinesis

EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Instances (1/1) Info

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

Instance ID = i-0ae3d59cab136daf0

Clear filters

< 1 >

☒	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☒	TestEC2	i-0ae3d59cab136daf0	Running	t2.micro	-	View alarms	us-west-2b

Instance: i-0ae3d59cab136daf0 (TestEC2)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

Instance summary

Instance ID

i-0ae3d59cab136daf0 (TestEC2)

IPv6 address

-

Hostname type

IP name: ip-172-31-22-123.us-west-2.compute.internal

Public IPv4 address

-

Instance state

Running

Private IP DNS name (IPv4 only)

ip-172-31-22-123.us-west-2.compute.internal

Private IPv4 addresses

172.31.22.123

IPv4 DNS

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences