

## Step 1: Deploy Scaleway VPN Gateway to get its public IP

```
resource "scaleway_s2s_vpn_gateway" "vgw" {
  name          = "workshop-vpn-gw"
  private_network_id = scaleway_vpc_private_network.pn.id
  gateway_type    = "VGW-S"
}
```

## Step 2: Deploy AWS Customer Gateway

Name	Customer gateway ID	State	BGP ASN	IP address	Type
scaleway	cgw-05f28ebf84cf6acb	Available	12876	163.172.175.212	ipsec.1

  

**Customer gateway cgw-05f28ebf84cf6acb / scaleway**

**Details** | Tags

<b>Customer gateway ID</b> cgw-05f28ebf84cf6acb	<b>State</b> Available	<b>Type</b> ipsec.1	<b>IP address</b> 163.172.175.212
<b>BGP ASN</b> 12876	<b>Certificate ARN</b> -	<b>Device</b> scaleway	

Put the public IP of the step 1 and the BGP ASN of Scaleway 12876

## Step 3: Deploy AWS VPG, nominate the ASN as 65000 or use Amazon's default ASN, attach it to VPC

Name	Virtual private gateway ID	State	VPC attachment state	Type	VPC
aws	vgw-058632ea477c14785	Available	Attached	ipsec.1	vpc-06f20dd971f2d4c8d

  

**Virtual private gateway vgw-058632ea477c14785 / aws**

**Details** | Tags

<b>Virtual private gateway ID</b> vgw-058632ea477c14785	<b>State</b> Available	<b>Type</b> ipsec.1	<b>VPC</b> vpc-06f20dd971f2d4c8d
<b>Amazon ASN</b> 65000	<b>VPC attachment state</b> Attached		

Step 4: Create a AWS Cloudwatch log group to collect ipsec and bgp logs, and create AWS S2S VPN Connection, link the VPG and Customer Gateway, no worries about the rest of the configuration

Find out the public IP address of Tunnel 1, here is 13.39.228.104

Download the configuration

VPN connections (1/1) <a href="#">Info</a>					
<a href="#">Actions</a> <a href="#">Download configuration</a> <a href="#">Create VPN connection</a>					
Name	VPN ID	State	Virtual private gateway	Transit gateway	VPN Concentrator
vpn	vpn-093906512c2297709	<span>Available</span>	vgw-058632ea477c14785	-	-

## Download configuration

Download a sample configuration based on your customer gateway. Note that this is a sample only, and that it will require modification for using Advanced Algorithms, Certificates, and/IPv6.

**Vendor**  
The manufacturer of the customer gateway device (for example, Cisco Systems, Inc).

Generic

**Platform**  
The class of the customer gateway device (for example, J-Series).

Generic

**Software**  
The operating system running on the customer gateway device (for example, ScreenOS).

Vendor Agnostic

**IKE version**  
The IKE version you are using for your VPN connection.

ikev1

**Include sample type - optional**

Enable

**Sample type**  
The default sample type compatibility mode includes all options. The recommended mode restricts options to only the most secure settings (IKEv2, etc.).

Select sample type

[Cancel](#) [Download](#)

Confirm all the setup and also the IP of BGP interface

#### #4: Border Gateway Protocol (BGP) Configuration:

The Border Gateway Protocol (BGPv4) is used within the tunnel, between the inside IP addresses, to exchange routes from the VPC to your home network. Each BGP router has an Autonomous System Number (ASN). Your ASN was provided to AWS when the Customer Gateway was created.

##### BGP Configuration Options:

- Customer Gateway ASN : 12876
- Virtual Private Gateway ASN : 65000
- Neighbor IP Address : 169.254.81.149
- Neighbor Hold Time : 30

Configure BGP to announce routes to the Virtual Private Gateway. The gateway will announce prefixes to your customer gateway based upon the prefix you assigned to the VPC at creation time.

#### Step 5: Deploy Scaleway Customer Gateway and build connection

```
resource "scaleway_s2s_vpn_customer_gateway" "cgw" {  
    name      = "workshop-customer-gw"  
    ipv4_public = var.cgw_ip  
    asn       = var.cgw_asn  
}  
  
resource "scaleway_s2s_vpn_routing_policy" "policy" {  
    name      = "workshop-vpn-policy"  
    is_ipv6   = false  
    prefix_filter_in = ["172.31.0.0/16"] #VPC of AWS  
    prefix_filter_out = ["172.16.64.0/22"] #VPC of SCW  
}  
  
resource "scaleway_s2s_vpn_connection" "main" {  
    name           = "workshop-connection"  
    vpn_gateway_id = scaleway_s2s_vpn_gateway.vgw.id  
    customer_gateway_id = scaleway_s2s_vpn_customer_gateway.cgw.id  
    initiation_policy = "customer_gateway"  
    enable_route_propagation = true  
  
    bgp_config_ipv4 {  
        routing_policy_id = scaleway_s2s_vpn_routing_policy.policy.id  
        private_ip       = "169.254.81.150/30" #Use another ip but within the same CIDR  
        peer_private_ip  = "169.254.81.149/30" #BGP interface ip used by AWS  
    }  
  
    ikev2_ciphers {  
        encryption = "aes256"  
        integrity  = "sha256"  
        dh_group   = "modp2048"  
    }  
}
```

```

esp_ciphers {
    encryption = "aes256"
    integrity  = "sha256"
    dh_group   = "modp2048"
}

}

data "scaleway_secret_version" "s2s_psk" {
    secret_id = scaleway_s2s_vpn_connection.main.secret_id
    revision  = tostring(scaleway_s2s_vpn_connection.main.secret_version)
}

output "psk" {
    value     = data.scaleway_secret_version.s2s_psk.data
    sensitive = true
}

```

Step 6: Fetch the psk secret of Scaleway's VPN connection and put it into AWS Tunnel 1  
jwang@mac workshop\_vpn % scw secret secret list region=fr-par -o json | jq .

```

[
{
    "id": "6382b71a-a958-40a3-952c-890c19354fa0",
    "project_id": "50fc0d26-96ad-4d8b-8b37-aeff2d19396a",
    "name": "connection_1fc87ea9-cb89-4aea-bceb-4a5f6a04c0ba",
    "status": "ready",
    "created_at": "2026-02-05T15:06:58.823909Z",
    "updated_at": "2026-02-05T15:06:58.823909Z",
    "tags": [
        "S2S VPN"
    ],
    "version_count": 1,
    "description": "Generated by Scaleway",
    "managed": false,
    "protected": true,
    "type": "opaque",
    "path": "/s2s_vpn",
    "ephemeral_policy": null,
    "used_by": [
        "s2s_vpn"
    ],
    "deletion_requested_at": null,
    "region": "fr-par"
}
]
```

scw secret version list 6382b71a-a958-40a3-952c-890c19354fa0 region=fr-par

Python

REVISION	SECRET ID	STATUS	CREATED AT
UPDATED AT	DELETED AT	DESCRIPTION	LATEST DELETION REQUESTED AT
1	6382b71a-a958-40a3-952c-890c19354fa0	enabled	25 minutes ago 25
minutes ago -	-	true	-

scw secret version access 6382b71a-a958-40a3-952c-890c19354fa0 revision=1 region=fr-par  
SecretID 6382b71a-a958-40a3-952c-890c19354fa0

Revision 1

Data QlpDZINKOU1uUThwMzNFWIBPcVp5TmpUR2xNSVZtZm4=  
Type opaque

```
echo 'QlpDZINKOU1uUThwMzNFWIBPcVp5TmpUR2xNSVZtZm4=' | base64 -D  
BZCfSJ9MnQ8p33EZPOqZyNjTGIMIVmf
```

Copy BZCfSJ9MnQ8p33EZPOqZyNjTGIMIVmf into tunnel 1, meanwhile make sure all the encryption rules are align with what we configured at Scaleway

We run \*\*IPsec over IKEv2\*\* on both sides and made sure everything lines up:

Parameter	Our AWS side	Our Scaleway side
IKE version	IKEv2	IKEv2
Phase 1 encrypt	AES-256	aes256
Phase 1 integrity	SHA2-256	sha256
Phase 1 DH	14 (MODP2048)	modp2048
Phase 2 encrypt	AES-256	aes256
Phase 2 integrity	SHA2-256	sha256
Phase 2 DH	14	modp2048
DPD	restart after 30 s	(aligned as needed)

**Inside IPv4 CIDR**  
A size /30 IPv4 CIDR block from the 169.254.0.0/16 range.

X

**Pre-shared key storage** | [Info](#)  
 Standard  
 Secrets Manager

**Pre-shared key**  
The pre-shared key must have 8-64 characters. Valid characters: A-Z, a-z, 0-9, \_ and . The key cannot begin with a zero.

**Phase 1 encryption algorithms**  
The permitted encryption algorithms for the VPN tunnel for phase 1 IKE negotiations.  
 ▼  
AES256 X

**Phase 2 encryption algorithms**  
The permitted encryption algorithms for the VPN tunnel for phase 2 IKE negotiations.  
 ▼  
AES256 X

**Phase 1 integrity algorithms**  
The permitted integrity algorithms for the VPN tunnel for phase 1 IKE negotiations.  
 ▼  
SHA2-256 X

**Phase 2 integrity algorithms**  
The permitted integrity algorithms for the VPN tunnel for phase 2 IKE negotiations.  
 ▼  
SHA2-256 X

**Phase 1 DH group numbers**  
The permitted Diffie-Hellman group numbers for the VPN tunnel for phase 1 IKE negotiations.  
 ▼  
14 X

**Phase 2 DH group numbers**  
The permitted Diffie-Hellman group numbers for the VPN tunnel for phase 2 IKE negotiations.  
 ▼  
14 X

**IKE Version**  
The internet key exchange (IKE) version permitted for the VPN tunnel.  
 ▼  
ikev2 X

Ask AWS to start the session and when it's timeout do restart

**DPD timeout (seconds)**  
The number of seconds after which a DPD timeout occurs.  
 ▼  
 Supported values must be 30 or higher.

**DPD timeout action** | [Info](#)  
 Clear  
 Restart  
 None

**Startup action** | [Info](#)  
 Add  
 Start

**Tunnel activity log**  
Tunnel activity log captures log messages for IPsec activity and DPD protocol messages.  
 Enable

**Destination**  
 Send to CloudWatch logs

**Amazon CloudWatch log group**  
 C

**Output format**  
 JSON  
 Text

**Tunnel BGP log - new**  
Tunnel BGP log captures log messages for BGP activity.  
 Enable

**Destination**  
 Send to CloudWatch logs

**Amazon CloudWatch log group**  
 C

**Output format**  
 JSON  
 Text

**Tunnel maintenance**  
**Tunnel endpoint lifecycle control** | [Info](#)  
Tunnel endpoint lifecycle control provides control over the schedule of endpoint replacements.  
 Turn on

▼ Tunnel 1 options <a href="#">Info</a>		Phase 1 integrity algorithms SHA2-256		Phase 1 DH group numbers 14		Phase 1 lifetime 28800	
Phase 1 encryption algorithms AES256		Phase 2 integrity algorithms SHA2-256		Phase 2 DH group numbers 14		Phase 2 lifetime 3600	
IKE version ikev2		Rekey fuzz 100		DPD timeout 30		Startup action start	
Rekey margin time 270		Replay window size 1024		DPD timeout action restart		Tunnel endpoint lifecycle control Off	
CloudWatch log group for tunnel VPN log <a href="#">vpn</a>		Output format for tunnel VPN log text		Tunnel endpoint lifecycle control Off		Tunnel VPN log <a href="#">Enabled</a>	
CloudWatch log group for tunnel BGP log <a href="#">vpn</a>		Output format for tunnel BGP log text		Tunnel BGP log <a href="#">Enabled</a>			

## Step 7: Verify ipsec connection and BGP session

AWS:

Checkout Cloudwatch:

- ▶ 2026-02-05T16:13:03.524Z 1770307983524 2026-02-05 16:13:03.524Z sending packet: from 13.39.228.104 [UDP 4500] to cgw-05f28ebf84cf6acb [UDP 4500] (80 bytes) true true established established
- ▶ 2026-02-05T16:13:03.526Z 1770307983526 2026-02-05 16:13:03.526Z received packet: from cgw-05f28ebf84cf6acb [UDP 4500] to 13.39.228.104 [UDP 4500] (80 bytes) true true established established

Checkout Console

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

Find resource by attribute or tag

Name	VPN ID	State	Virtual private gateway	Transit gateway	VPN Concentrator ID	Customer gateway	Customer gateway add...	Inside IP version	Type
vpn	vpn-093906512c2297709	Available	vpe-058632ea477c14785	-	-	cgw-05f28ebf84cf6acb	163.172.175.212	IPv4	ips

VPN connection vpn-093906512c2297709 / vpn

Details [Tunnel details](#) Tags

⚠ This VPN connection is not using both tunnels. This mode of operation is not highly available and we strongly recommend you configure your second tunnel.

Tunnel state	Outside IP address	Inside IPv4 CIDR	Inside IPv6 CIDR	Status	Provisioning status	Last status change	Details	Certificate ARN
Tunnel 1	13.39.228.104	169.254.81.148/30	-	Up	Available	February 5, 2026, 16:42:35 (UTC+01:00)	1 BGP ROUTES	-
Tunnel 2	51.44.201.116	169.254.237.40/30	-	Down	Available	February 5, 2026, 15:56:05 (UTC+01:00)	IPSEC IS DOWN	-

▶ Tunnel 1 options [Info](#)

▶ Tunnel 2 options [Info](#)

Scaleway:

```
Python
scw s2s-vpn connection list

ID                                PROJECT ID
ORGANIZATION ID                   NAME          TAGS   CREATED AT
1fc87ea9-cb89-4aea-bceb-4a5f6a04c0ba 50fc0d26-96ad-4d8b-8b37-aeff2d19396a
50fc0d26-96ad-4d8b-8b37-aeff2d19396a workshop-connection []    43 minutes
ago
```

```
Python
scw s2s-vpn connection get 1fc87ea9-cb89-4aea-bceb-4a5f6a04c0ba
```

ID	1fc87ea9-cb89-4aea-bceb-4a5f6a04c0ba
ProjectID	50fc0d26-96ad-4d8b-8b37-aeff2d19396a
OrganizationID	50fc0d26-96ad-4d8b-8b37-aeff2d19396a
Name	workshop-connection
CreatedAt	44 minutes ago
UpdatedAt	9 minutes ago
Status	active
IsIPv6	false
InitiationPolicy	customer_gateway
SecretID	6382b71a-a958-40a3-952c-890c19354fa0
SecretRevision	1
Ikev2Ciphers.0.Encryption	aes256
Ikev2Ciphers.0.Integrity	sha256
Ikev2Ciphers.0.DhGroup	modp2048
EspCiphers.0.Encryption	aes256
EspCiphers.0.Integrity	sha256
EspCiphers.0.DhGroup	modp2048
RoutePropagationEnabled	true
VpnGatewayID	e25edab1-3f7c-43d9-ac15-29548d08618d
CustomerGatewayID	0eb9471e-4e35-4a71-9614-cb8715542856
<b>TunnelStatus</b>	<b>up</b>
TunnelStatusIPv4	unknown_tunnel_status
TunnelStatusIPv6	unknown_tunnel_status
<b>BgpStatusIPv4</b>	<b>up</b>
BgpStatusIPv6	disabled
BgpSessionIPv4.RoutingPolicyID	3ca3c7b5-b81d-4514-96a0-aaf103931e88
BgpSessionIPv4.PrivateIP	169.254.81.150/30
BgpSessionIPv4.PeerPrivateIP	169.254.81.149/30
Region	fr-par

Need to make sure Tunnel and BGP are both UP