Google Cloud VPN Interop Guide

Using Cloud VPN with Amazon Web Services (AWS)™ Virtual Private Gateway

Disclaimer: This interoperability guide is intended to be informational in nature and includes examples only. Customers should verify this information via testing.

Contents

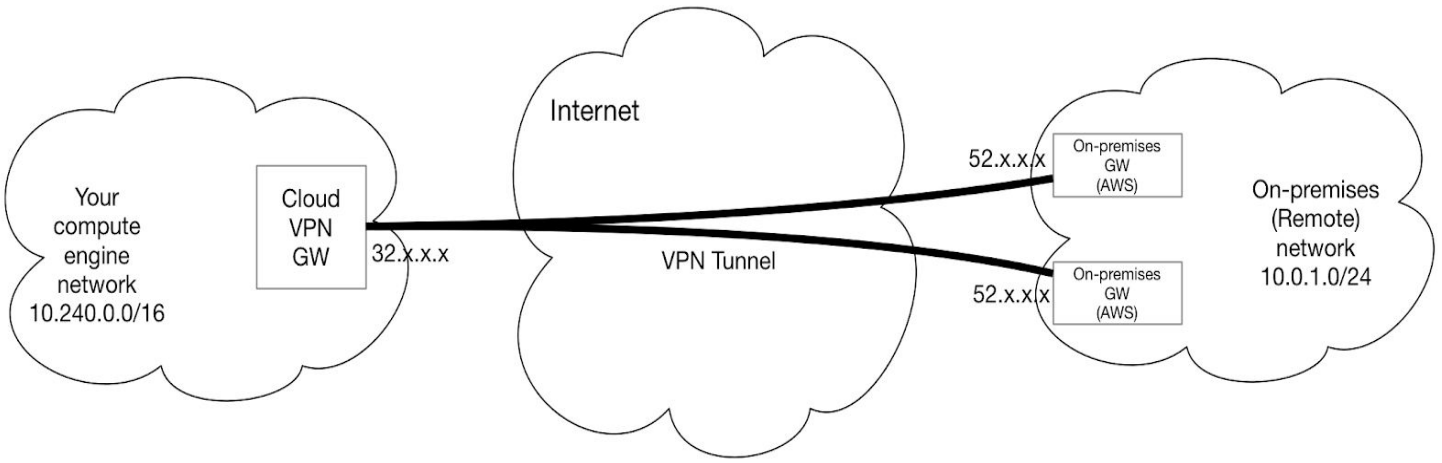
Introduction

This guide walks you through the process of configuring the AWS Virtual Private Gateway for integration with Google Cloud VPN. This information is provided as an example only. If you are using this guide to configure your AWS implementation, substitute the correct IP information for your environment.

Topology

This guide describes the following VPN topologies:

- A site-to-site Route-based IPsec VPN tunnel configuration.
- A site-to-site IPsec VPN tunnel configuration using Google Cloud Router and dynamic routing with the BGP protocol.



Preparation

Overview

NOTE: The configuration samples in this guide include numerous value substitutions that are provided only as examples. For any references to IP addresses, device IDs, shared secrets, keys, account information, or project names, replace the given values with the appropriate values for your environment.

This guide assists you in the creation of IPsec connectivity from AWS to Google Cloud. The following is a high-level overview of the configuration process:

1. Configure the Amazon Virtual Private Gateway.
2. Configure the Amazon Customer Gateway.
3. Configure the Google Cloud Platform VPN.
4. Set up the VPN connection.
5. Connect to GCP.
6. Test the tunnel.

Getting started

The first step is to establish the base networking environment in AWS, which is called Virtual Private Cloud (VPC). Amazon provides documentation for getting started with AWS networking. The basic concepts to understand are:

- **Virtual Private Cloud** – a customer-defined private network space in AWS.
- **Virtual Private Gateway** – the VPN concentrator on the Amazon side of the VPN connection.
- **Customer Gateway** – an AWS reference to the remote IPsec endpoint. In this case, the Google Cloud Platform (GCP) VPN gateway.

IPsec parameters

This table covers the IPsec parameters to use when configuring VPN gateways and tunnels as described in this document. The IPsec connectivity covered in this guide uses the **pre-shared key** generated by AWS for authentication. AWS supports only IKEv1. For more detail, see this information about GCP-supported IKEv1 ciphers.

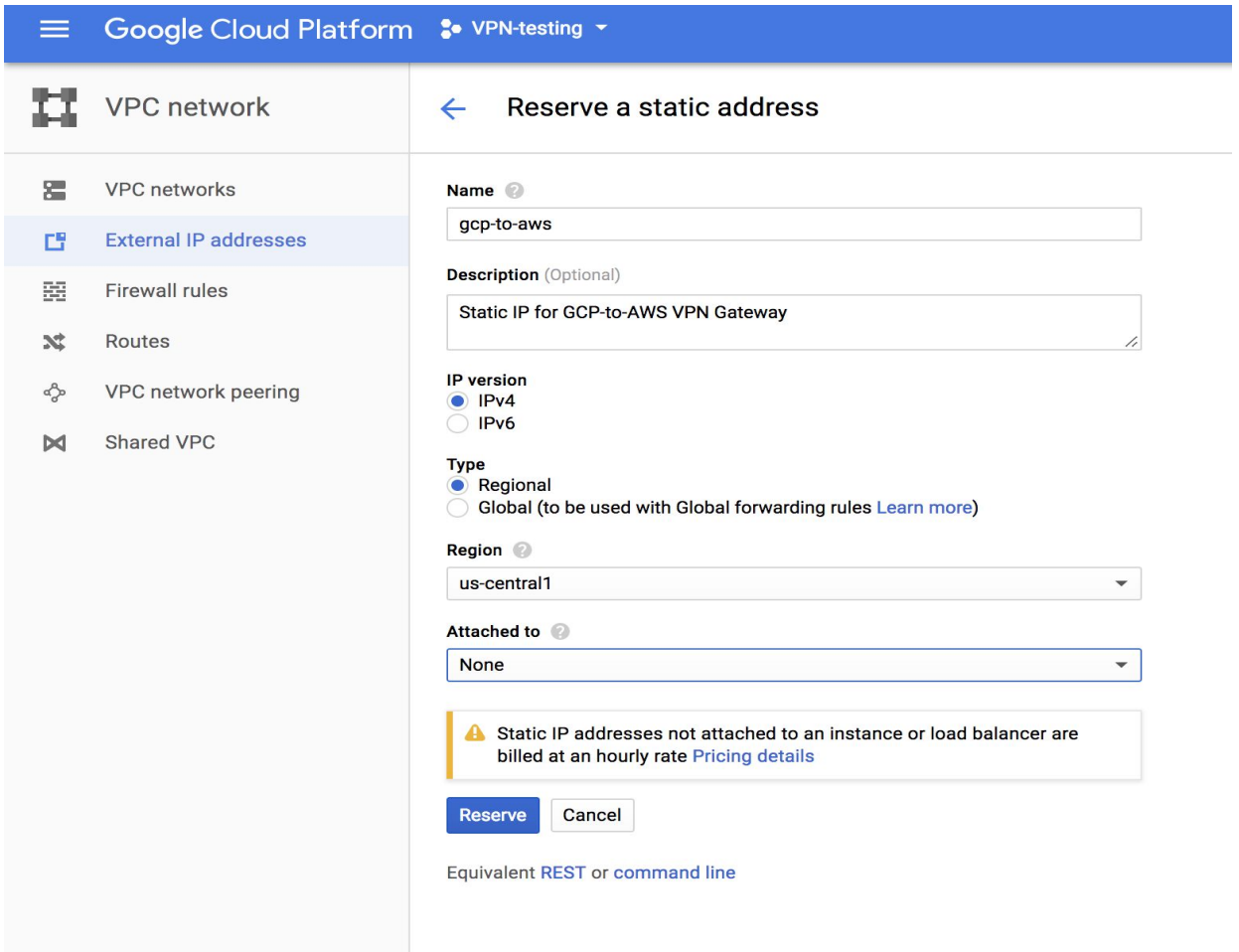
Parameter	Value
IPsec Mode	ESP+Auth Tunnel mode (Site-to-Site)
Authentication Protocol	Pre-shared Key
Key Exchange	IKEv1

Perfect Forward Secrecy (PFS)	on

IPsec VPN using static routes

Reserve an external static IP address for GCP

The AWS VPN configuration requires a remote VPN gateway IP address in advance. In the GCP console, reserve a static external IP address by selecting the **External IP addresses** option under the [VPC networks menu]() option. This is shown in the following screenshot.

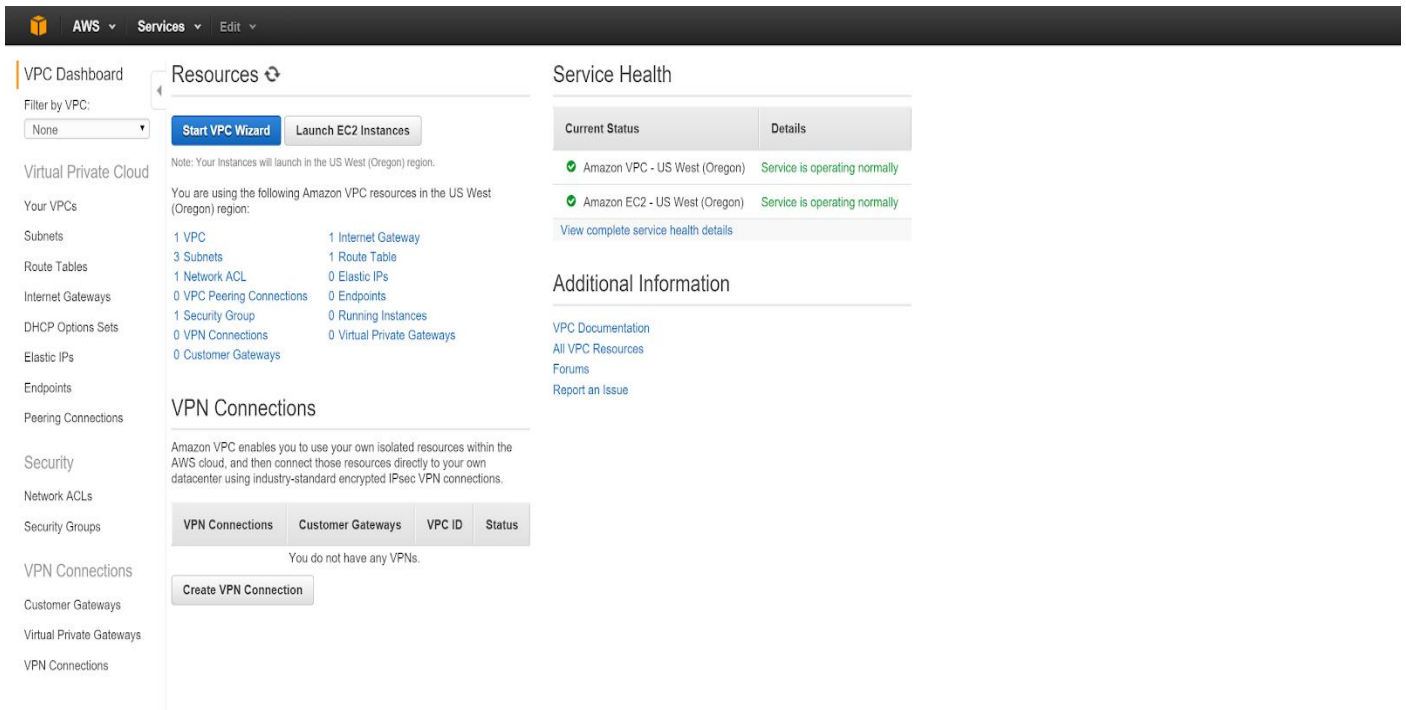


Configuration - AWS

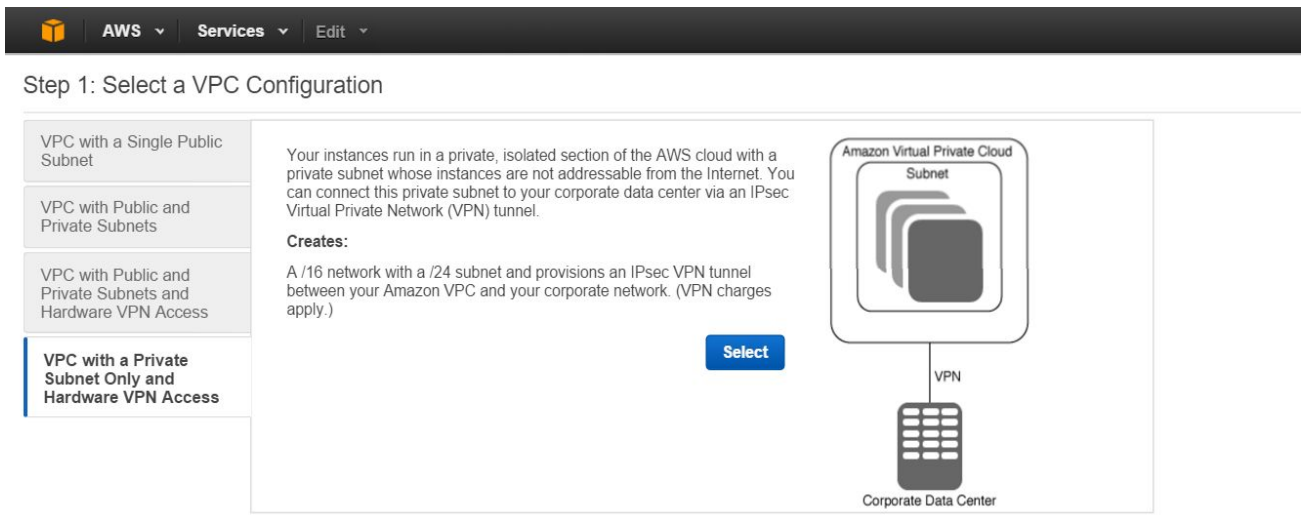
For this exercise, create a VPC network and subnet configuration using the AWS **VPC Wizard** to connect to Google Cloud Platform. The VPC Wizard steps through the creation and configuration of a new VPC network.

Creating the AWS VPC Network

1. Sign in to the AWS Management Console and select **VPC** from the main services menu. New AWS accounts all have a default VPC.



2. Select an IP subnet topology. There are options for various combinations of private and public IP addressing, with or without VPN connectivity. Once you select a topology and configuration, you cannot change it. For this test environment, select **Private Subnet Only VPC with Hardware VPN Access**:



3. Configure the VPC settings:

The following settings are required:

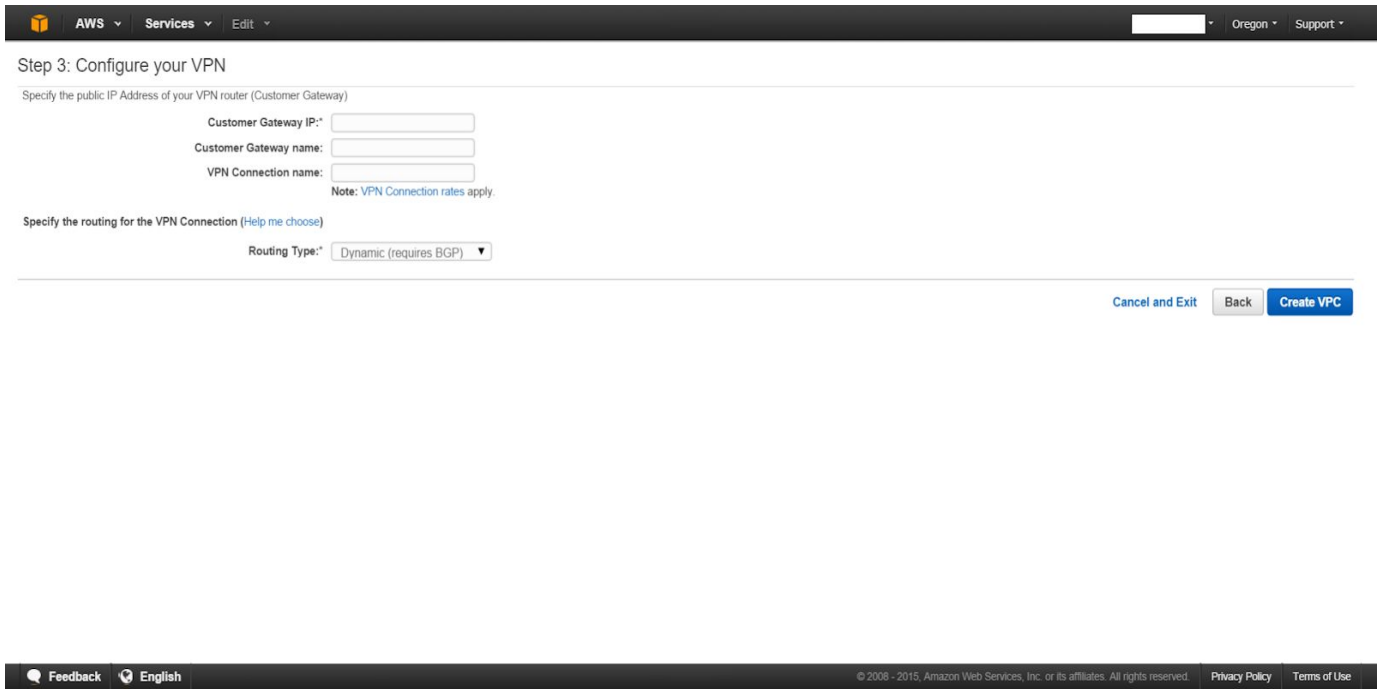
- **IP CIDR Block**: The CIDR block for the VPC network. Once you set this value, it cannot be changed. For this test configuration, enter **10.0.0.0/16**.
- **VPC Name**: The name of the VPC network. For this example, enter **GCP-Test**.

- **Private Subnet**: The first subnet allocated from the private IP CIDR block used for AWS services, including Amazon EC2. Enter **10.0.1.0/24,** which is the network on the AWS side that you want to connect to GCP.
- **Availability Zone**: The AWS Availability Zone into which the VPC is deployed. Leave this set to **no preference.**
- **Private Subnet Name**: A friendly name for the private subnet. Set this to **AWS-VPC**.
- **S3 Endpoint (not required)**: Amazon EC2 to Amazon S3 connectivity requires a public network link. This option deploys an Amazon S3 API gateway endpoint into the selected private subnet. This exercise does not require an Amazon S3 endpoint.
- **Enable DNS Hostnames**: Enables an automatic DNS hostname assignment through DHCP for the private subnet. Leave DNS hostnames enabled.
- **Hardware Tenancy**: Allows you to select a dedicated instance type for the VPN gateway. Use the default option.

4. When you complete the form, click **Next**.

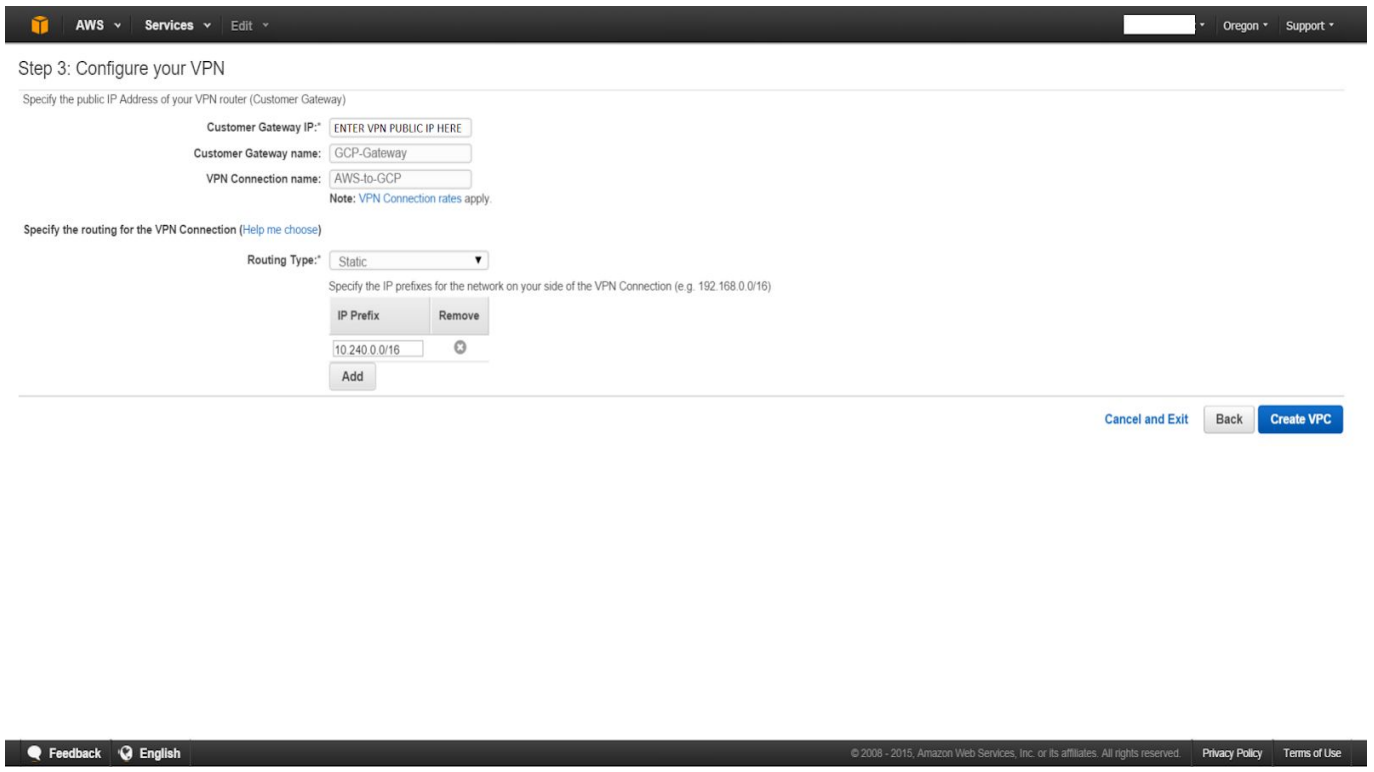
Configuring the AWS VPN

1. Enter the reserved GCP external IP address in the **Customer Gateway IP** field.

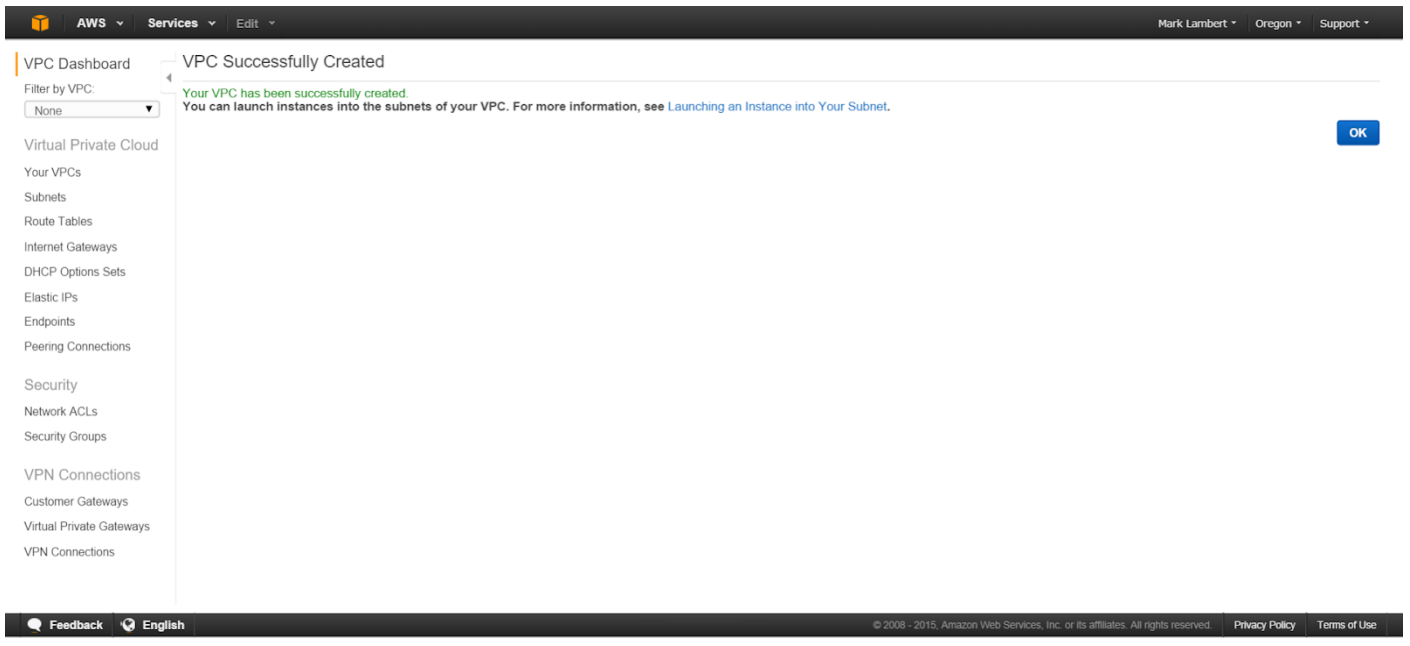


2. In addition to the Customer Gateway IP, enter a **Customer Gateway name** and a **VPN Connection name**.

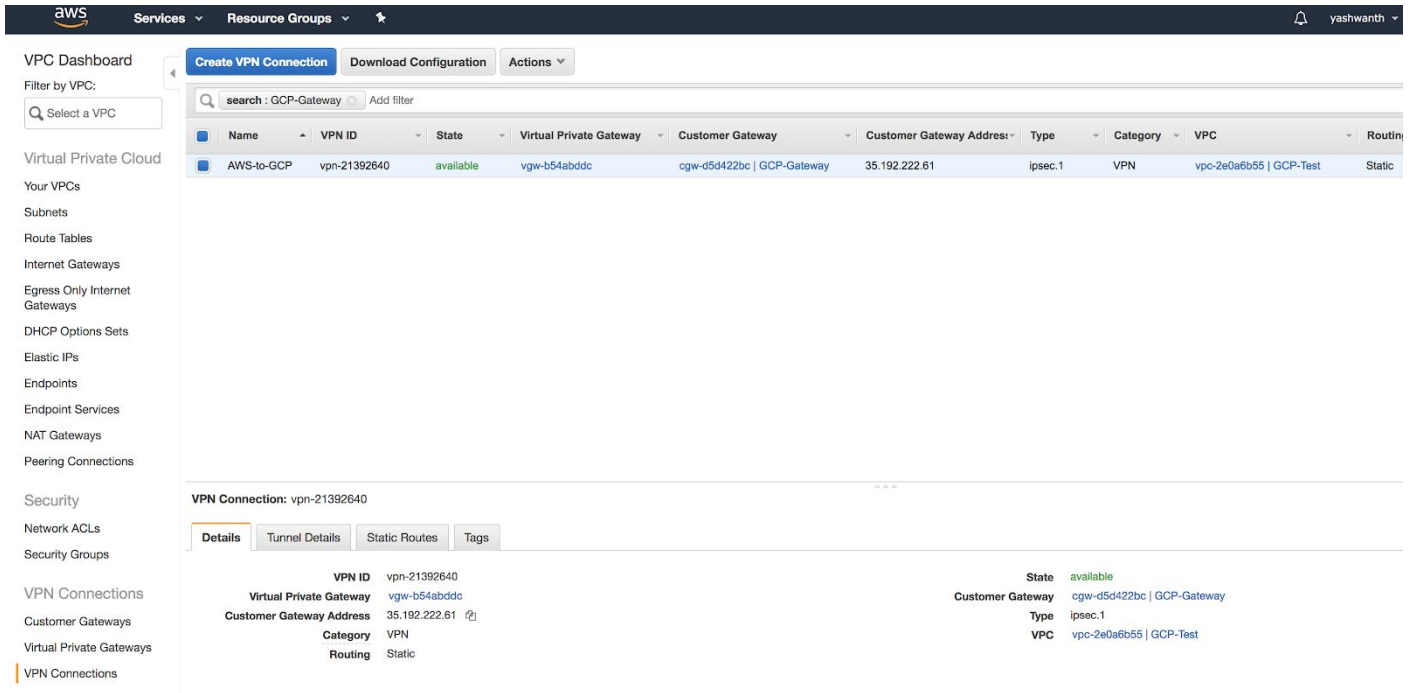
3. Choose a **Routing Type** for the VPN connection. This configuration uses a **Static route** type of VPN, so select **Static**. Enter the Google Cloud Platform subnet CIDR block under **IP Prefix**, and then click **Add**:



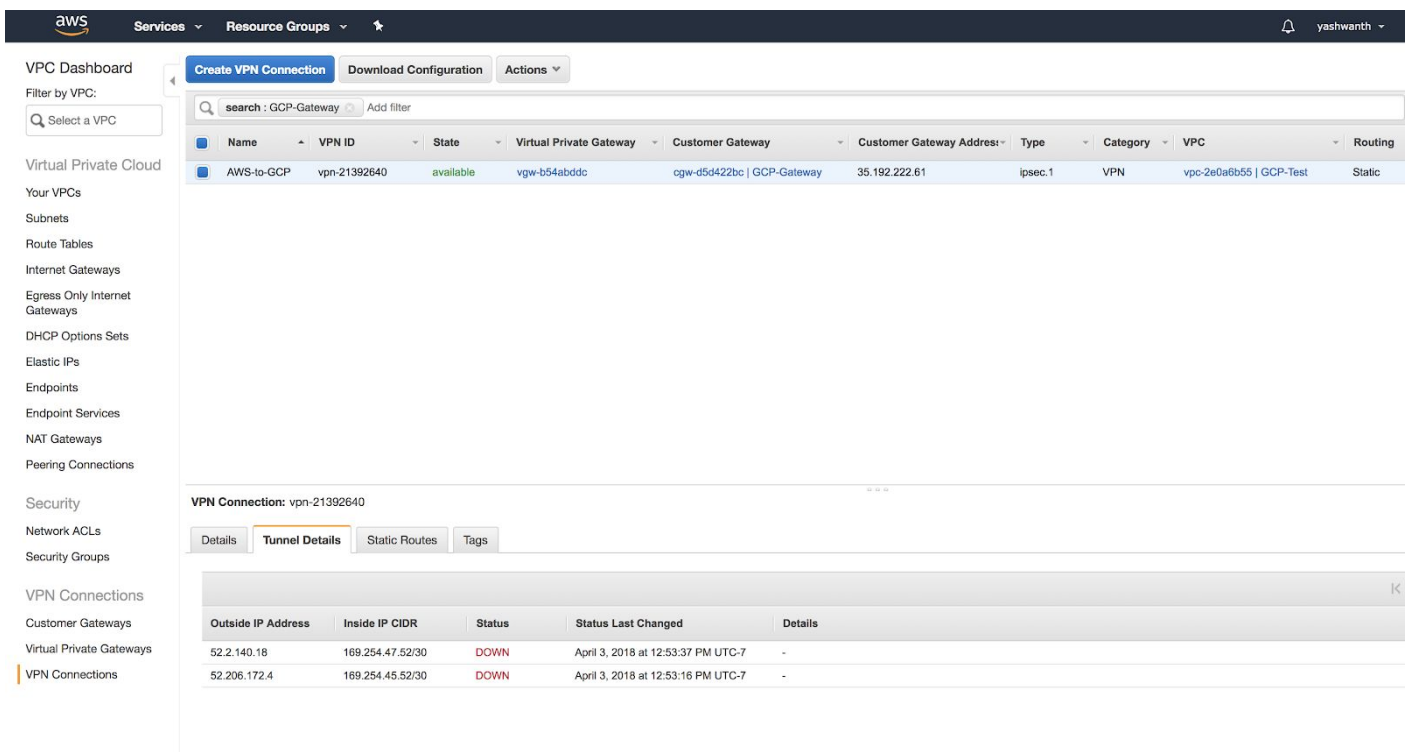
4. When all required configuration is completed, click **Create VPC** to create the new VPC and finish the wizard. VPC creation takes a minute or two to complete, after which the management console status is updated to show successful creation of the VPC.

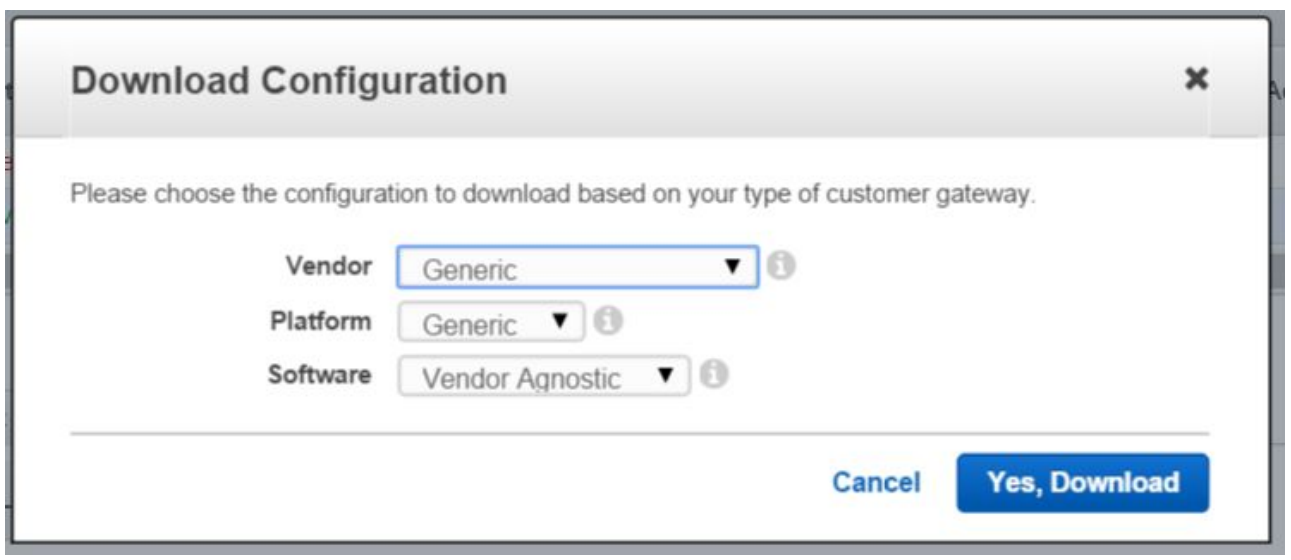


5. Select the newly created VPC from the VPC Dashboard in order to collect the configuration detail required to complete the GCP configuration:



6. Collect the IP addresses of the Amazon Virtual Private Gateway and the pre-shared keys used for IKE authentication that are automatically generated by AWS. This information is stored in the *configuration file*, which you can download by clicking **Download Configuration**. Although several device-specific options are available for the file configuration format, for GCP, select **Generic**:





The configuration file is an ASCII text file. Within the file, the auto-generated pre-shared key is listed under **Pre-Shared Key**.

A sample configuration file is provided below for reference.

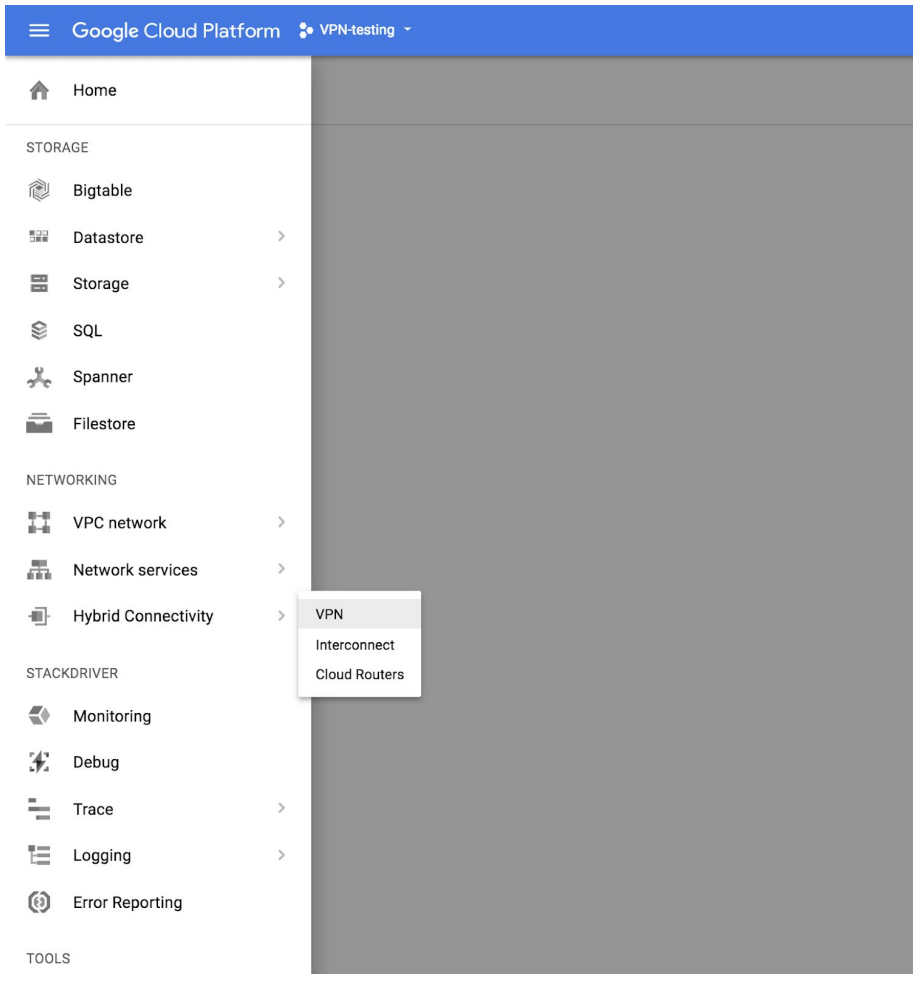
NOTE: AWS creates two VPN tunnels under “VPN connections,” and there are two sets of VPN parameters listed in the sample configuration file, one set for each tunnel. These parameters must match the tunnel parameters on the GCP side that you will configure later in this document:

```
Amazon Web Services
Virtual Private Cloud

VPN Connection Configuration
================================================================================
AWS utilizes unique identifiers to manipulate the configuration of
a VPN Connection. Each VPN Connection is assigned a VPN Connection Identifier
and is associated with two other identifiers, namely the
Customer Gateway Identifier and the Virtual Private Gateway Identifier.

Your VPN Connection ID                      : vpn-c1c6d9d3
Your Virtual Private Gateway ID          : vgw-f670afe8
Your Customer Gateway ID             : cgw-3548972b

A VPN Connection consists of a pair of IPSec tunnel security associations (SAs).
It is important that both tunnel security associations be configured.

IPSec Tunnel #1
================================================================================
#1: Internet Key Exchange Configuration

Configure the IKE SA as follows
  - Authentication Method    : Pre-Shared Key
  - Pre-Shared Key           : auto-generated-pre-shared-key
  - Authentication Algorithm : sha1
  - Encryption Algorithm     : aes-128-cbc
  - Lifetime                 : 28800 seconds
  - Phase 1 Negotiation Mode : main
  - Perfect Forward Secrecy  : Diffie-Hellman Group 2
```

Configuration - GCP Console

In the GCP Console, either select the GCP project into which the VPN will be deployed, or create a project. See more information on creating and managing projects.

1. To create a VPN, open the main services menu located at the top left corner in the console. Under **Networking**, select **Hybrid Connectivity** and **VPN**:

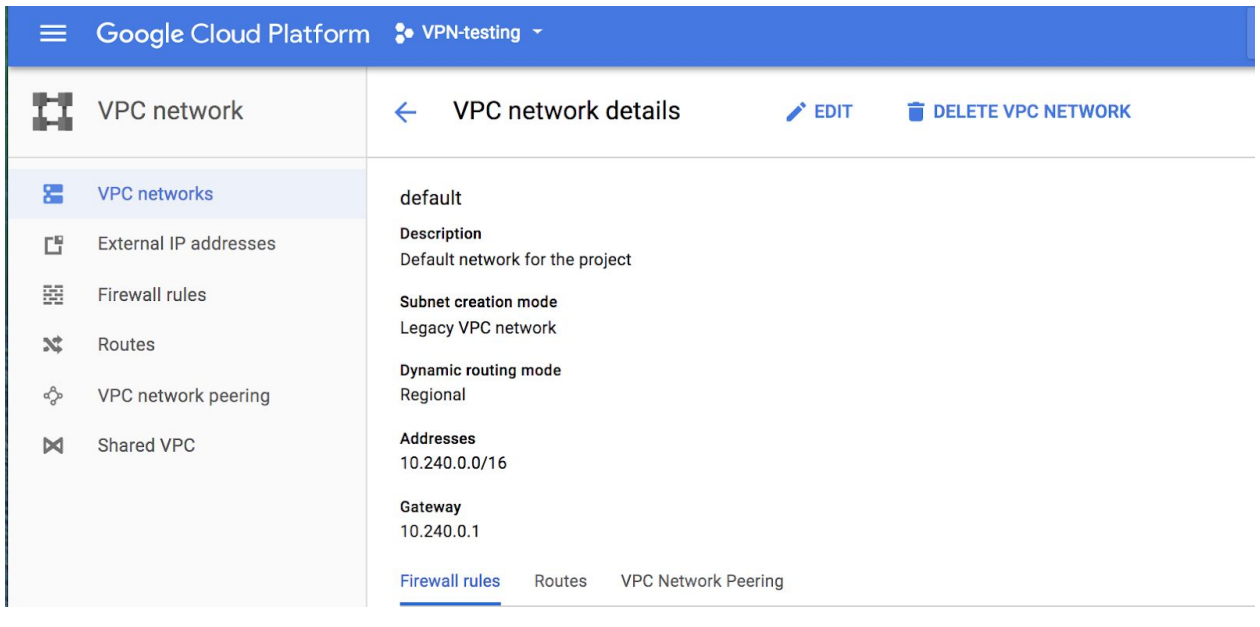


In GCP, all projects start with a single auto mode network named *default* at the time of project creation. This default network is configured with a private IP address space and a set of base firewall rules. This network provides a sufficient starting point for creating a site-to-site IPsec VPN as long as the CIDR address range on the AWS side doesn't overlap the GCP address range. More information on networking within the Google Cloud Platform can be found in the Networking section of the Google Cloud Platform documentation.

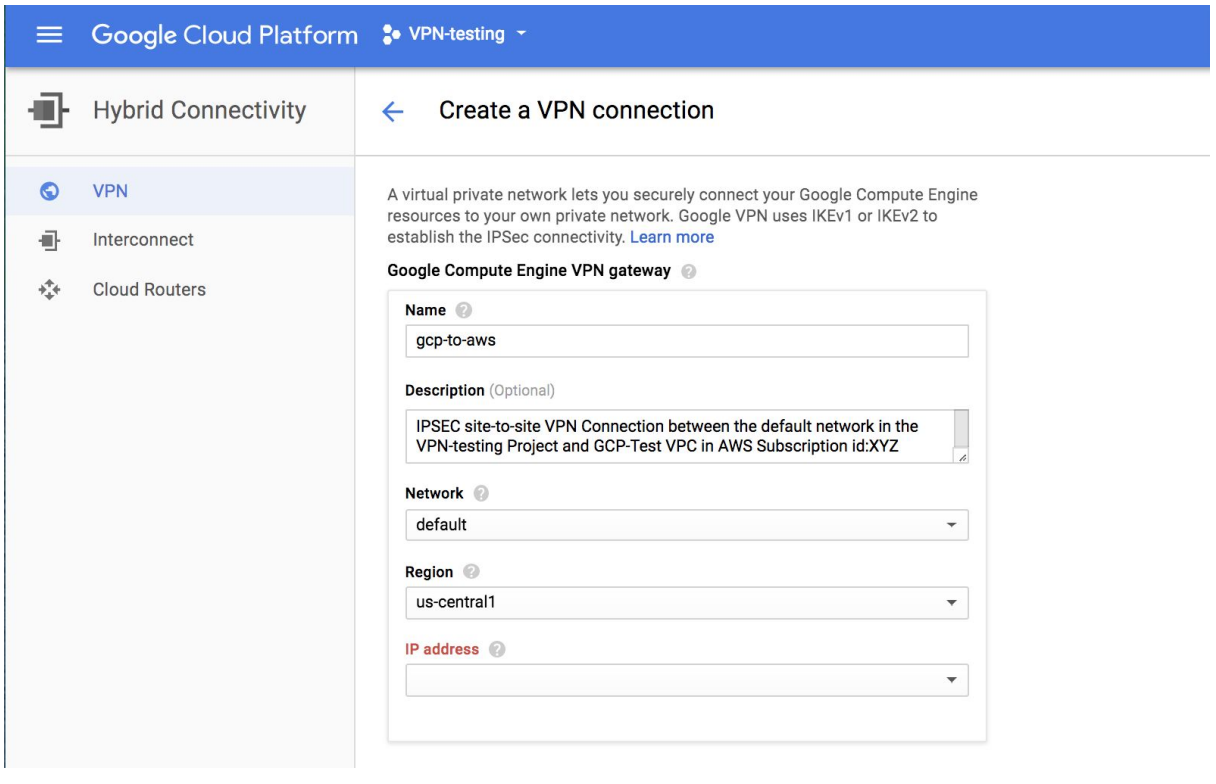
2. To configure the AWS side of the VPN, get the following two values from GCP:

   - **Customer Gateway IP Address**: the public IP address of the VPN gateway in Google Cloud.
   - **Routing Type/IP Prefix**: the private IP address space associated with the GCP Network.

   The address space is shown in the GCP console network overview. For this example configuration, the address space is 10.240.0.0/16:



3. To get the customer gateway IP address, create a Google Cloud VPN gateway. From the **Hybrid Connectivity** menu, select VPN and click **Create**:



4. Configure the following options for the GCP VPN gateway:

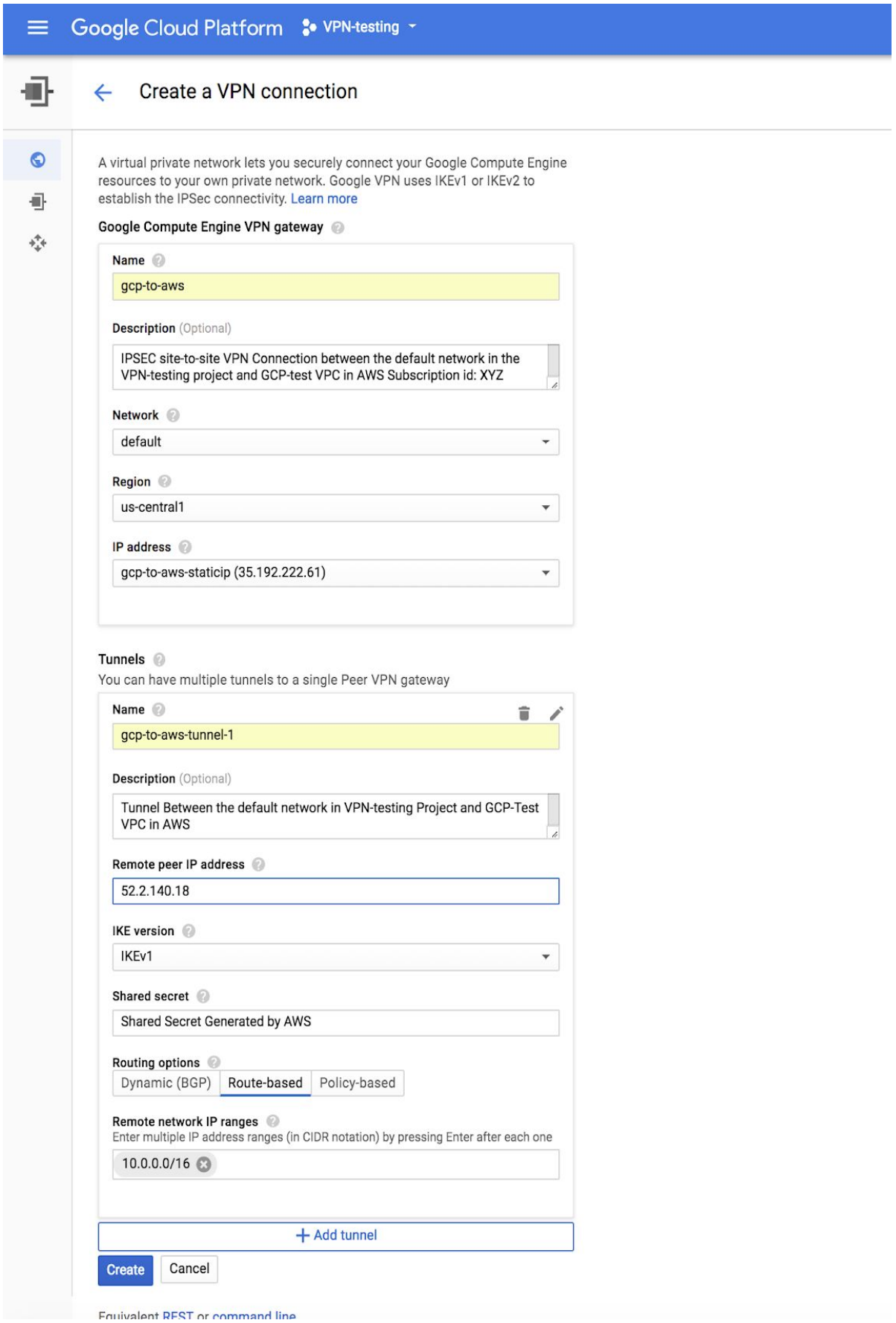
    - **Name**: a representative name for the VPN connection (must be lowercase).
    - **Description**: (optional) free form text describing the gateway.
    - **Network**: the VPC network to which the VPN gateway will be attached.
    - **Region**: the region into which the VPN gateway will be deployed.
    - **IP address**: a previously-reserved static public IP address to assign to the VPN gateway.

    a. Since each GCP VPN gateway can terminate multiple VPN tunnels, specify the parameters for each tunnel in the console fields.

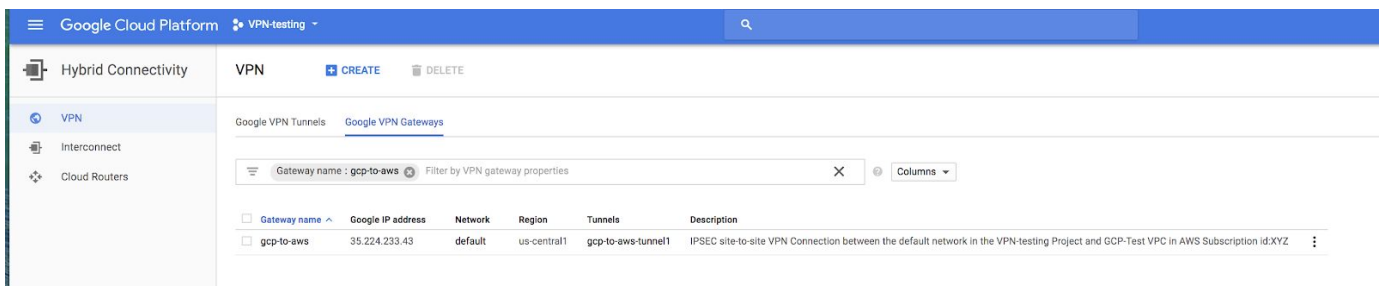
    b. Enter the AWS Virtual Private Gateway IP in the **Remote peer IP address** field and the pre-shared key in the **Shared Secret** field. Use the IP address collected from the Configuration - AWS section. Set the IKE version to IKEv1, since AWS is supports only this IKE version.

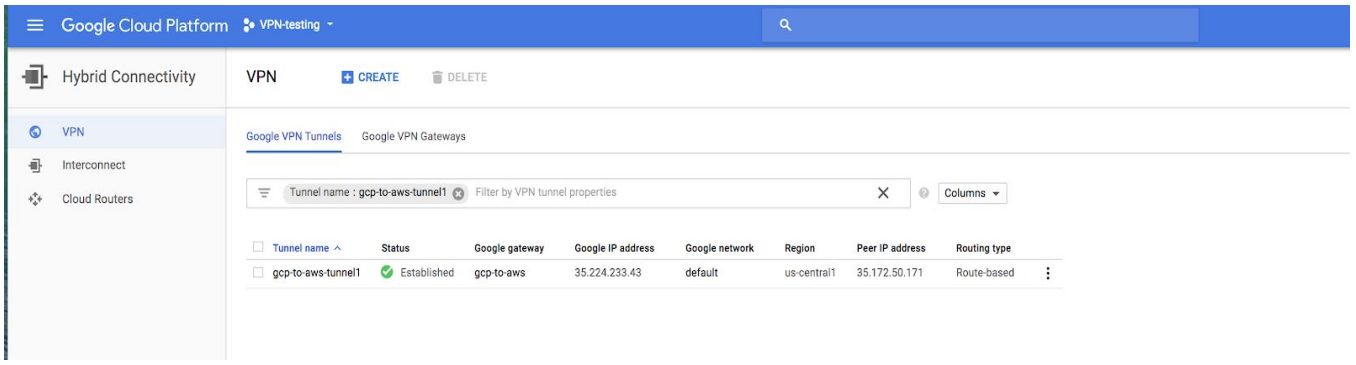
    c. Under the section **Routing Options**, select the **Route-based** tab, and enter the AWS network ranges as **Remote network IP ranges** . The Remote Network IP Ranges should include both the VPC CIDR block as well as any configured subnets.

    d. Since AWS requires two tunnels per VPN connection for redundancy, create an additional tunnel for the same GCP VPN gateway by clicking **Add Tunnel** to specify parameters for additional tunnels, including a different name and IP address than those used for Tunnel 1. Click **Create** to create the VPN gateway and tunnels that were specified.



5. Verify that the VPN gateway has been created and the connection to the AWS GCP-Test VPN has been established by looking at the tunnel status as shown below.





Configuration - GCP gcloud command-line tool

You can also configure Cloud VPN by using the gcloud command-line tool. Command-line configuration requires two steps. First, create the Cloud VPN gateway, and then create the tunnels used by the gateway.

Reserving an external static IP address

Reserve an external static IP address in the GCP network and region where the VPN gateway was created. Make a note of the address created for use in future steps.

```
gcloud compute addresses create vpn-static-ip --project vpn-testing /
    --region us-central1
```

Creating the Cloud VPN gateway

To create a Cloud VPN gateway, enter the following command:

```
gcloud compute target-vpn-gateways create gcp-to-aws /
    --network default --region us-central1
```

Creating forwarding rules

To create the three forwarding rules for the project's network forwarded through the gateway, enter the following commands.

Note: The GCP console creates these rules automatically.

```
gcloud compute forwarding-rules create gcp-to-aws-rule-udp4500 /
    --address gcp-static-ip  --ip-protocol UDP --ports 4500 /
    --region us-central1 --target-vpn-gateway gcp-to-aws

gcloud compute forwarding-rules create gcp-to-aws-rule-udp500 /
    --address gcp-static-ip --ip-protocol UDP --ports 500 /
    --region us-central1 --target-vpn-gateway gcp-to-aws

gcloud compute forwarding-rules create gcp-to-aws-rule-esp /
    --address gcp-static-ip --ip-protocol ESP /
    --region us-central1 --target-vpn-gateway gcp-to-aws
```

Creating the VPN Tunnels

Because AWS requires two VPN tunnels for redundancy, enter the following command for each tunnel. For tunnel 2, change the peer-address to a second on-premises IP address and the name to another unique name.

```
gcloud compute vpn-tunnels create gcp-to-awstunnel1 /
    --peer-address on-prem-IP-1 --ike-version 1 /
    --shared-secret SharedSecretGeneratedbyAWS /
    --target-vpn-gateway gcp-to-aws /
    --local-traffic-selector gcp-CIDR /
    --remote-traffic-selector on-prem-CIDR
```

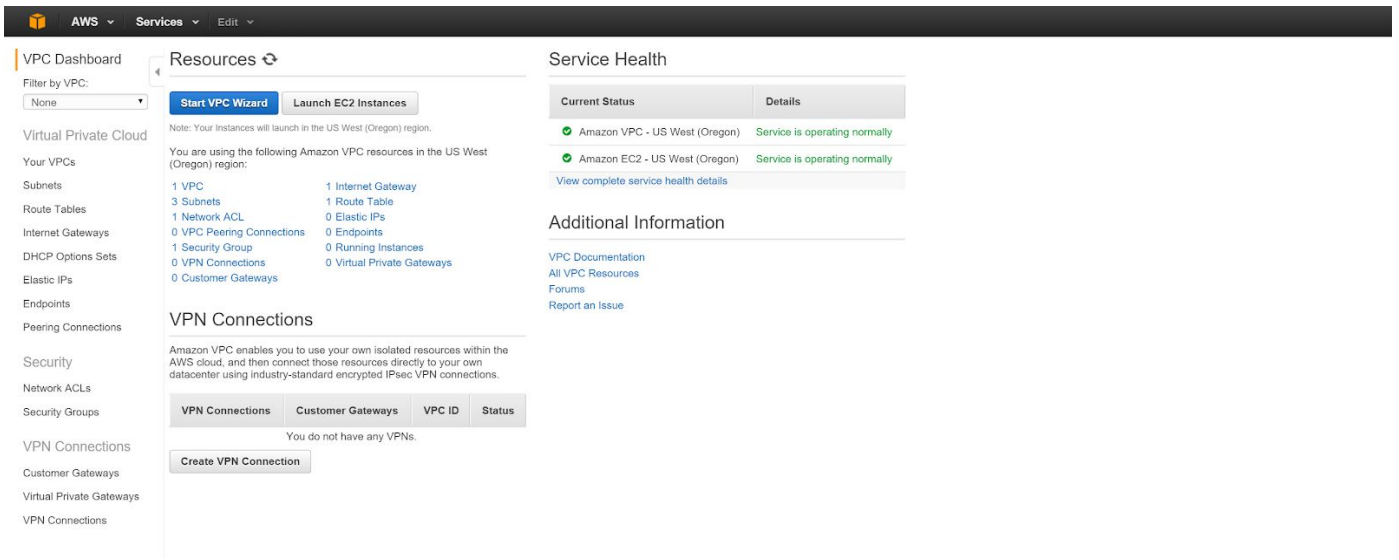
IPsec VPN using Cloud Router

Configuration - AWS

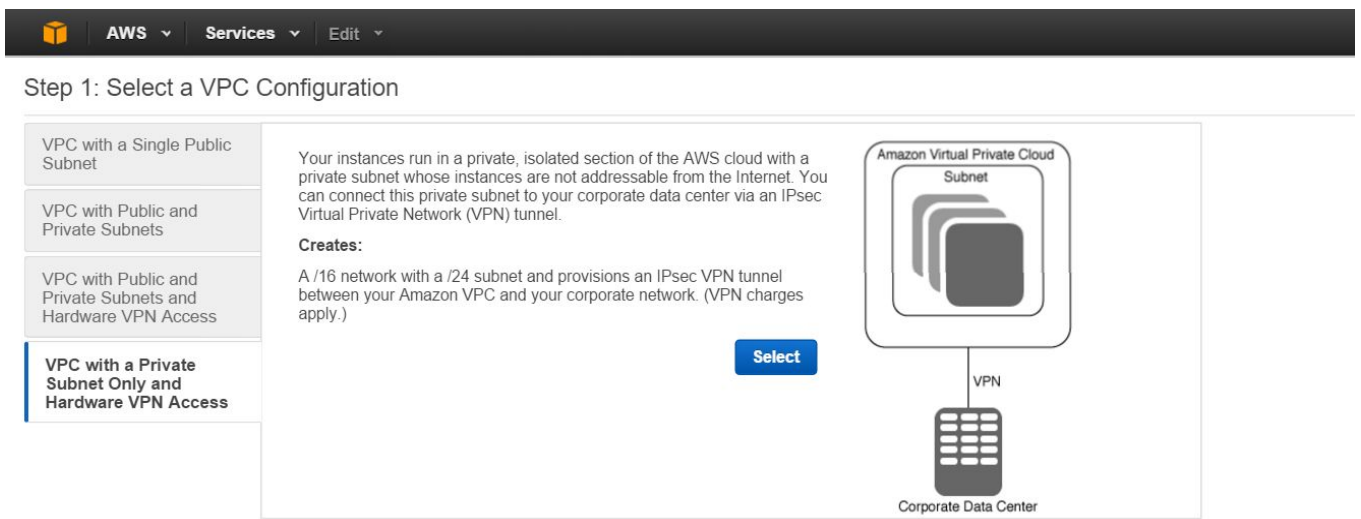
Creating the VPC network

Although new AWS accounts all have a default VPC network, for this exercise, create a new VPC network to connect to the Google Cloud Platform VPN gateway.

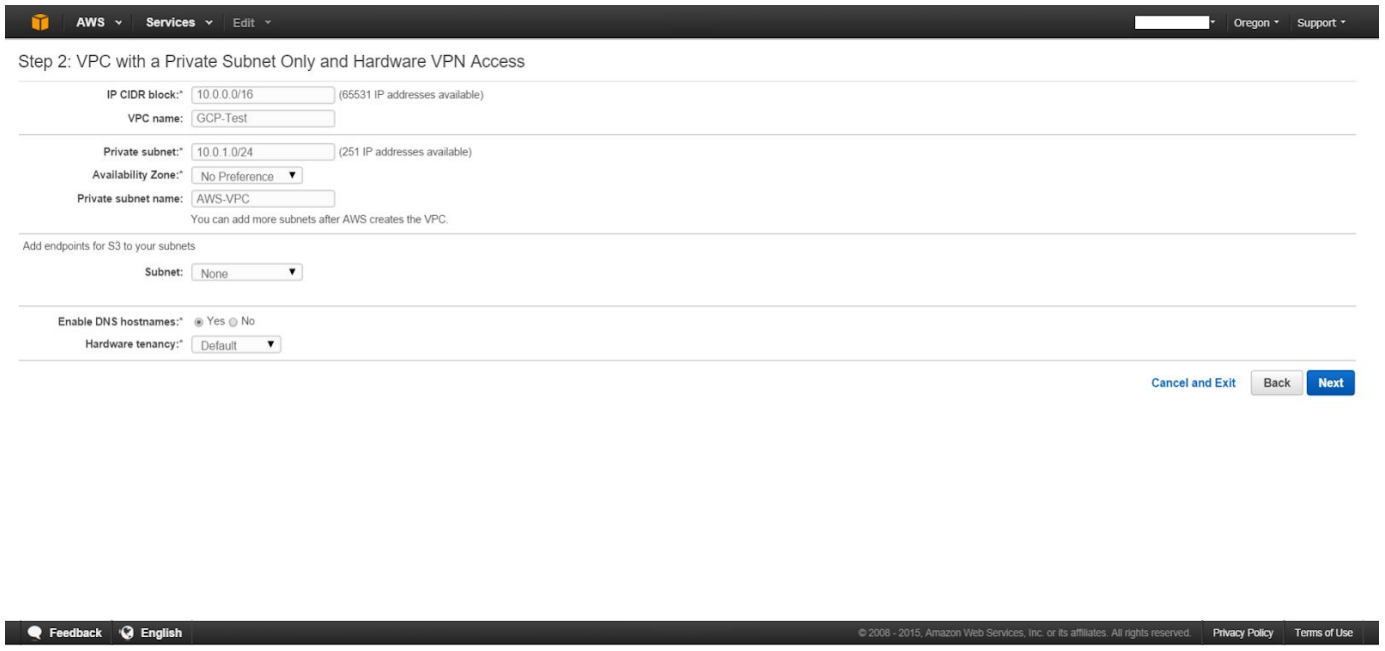
1. The VPC Wizard steps through the creation and configuration of a new VPC network. Using the **VPC Wizard**, sign in to the AWS Management Console and select *VPC* from the main services menu.



2. Select an IP subnet topology. There are options for various combinations of private and public IP addressing, with or without VPN connectivity. Once selected, the option cannot be changed. For the test environment, **Select** a **Private Subnet Only VPC with Hardware VPN Access**:



3. Configure the VPC settings:



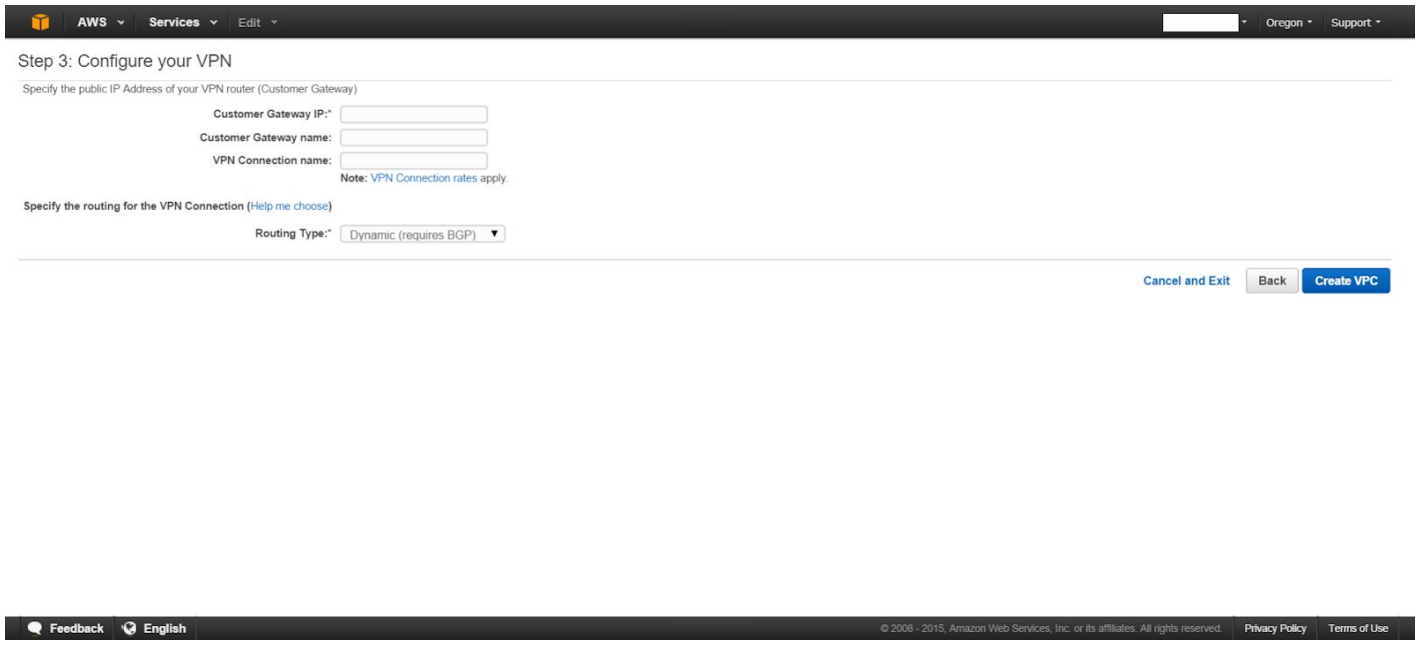
4. Configure the following required settings:

    - **IP CIDR Block**: The CIDR block for the VPC network. Once you set this value, you cannot change it. For this test, enter **10.0.0.0/16**.
    - **VPC Name**: The name of the VPC network. For this test, enter **GCP-Test**.
    - **Private Subnet**: The first subnet allocated from the private IP CIDR block used for AWS services, including Amazon EC2. Enter **10.0.1.0/24,** which is the network on the AWS side that you want to connect to GCP.
    - **Availability Zone**: The AWS Availability Zone into which the VPC network will be deployed. Leave this set to **no preference**.
    - **Private Subnet Name**: A friendly name for the private subnet. Set this to **AWS-VPC**.
    - **S3 Endpoint** (not required): EC2-to-S3 connectivity requires a public network link. This option deploys an Amazon S3 API gateway endpoint into the selected private subnet. This exercise does not require an Amazon S3 endpoint.
    - **Enable DNS Hostnames**: Enables automatic DNS hostname assignment by DHCP for the private subnet. Leave DNS hostnames enabled.
    - **Hardware Tenancy:** Allows selection of a dedicated instance type for the VPN gateway. Use the default option.

When you complete the form, click **Next**.

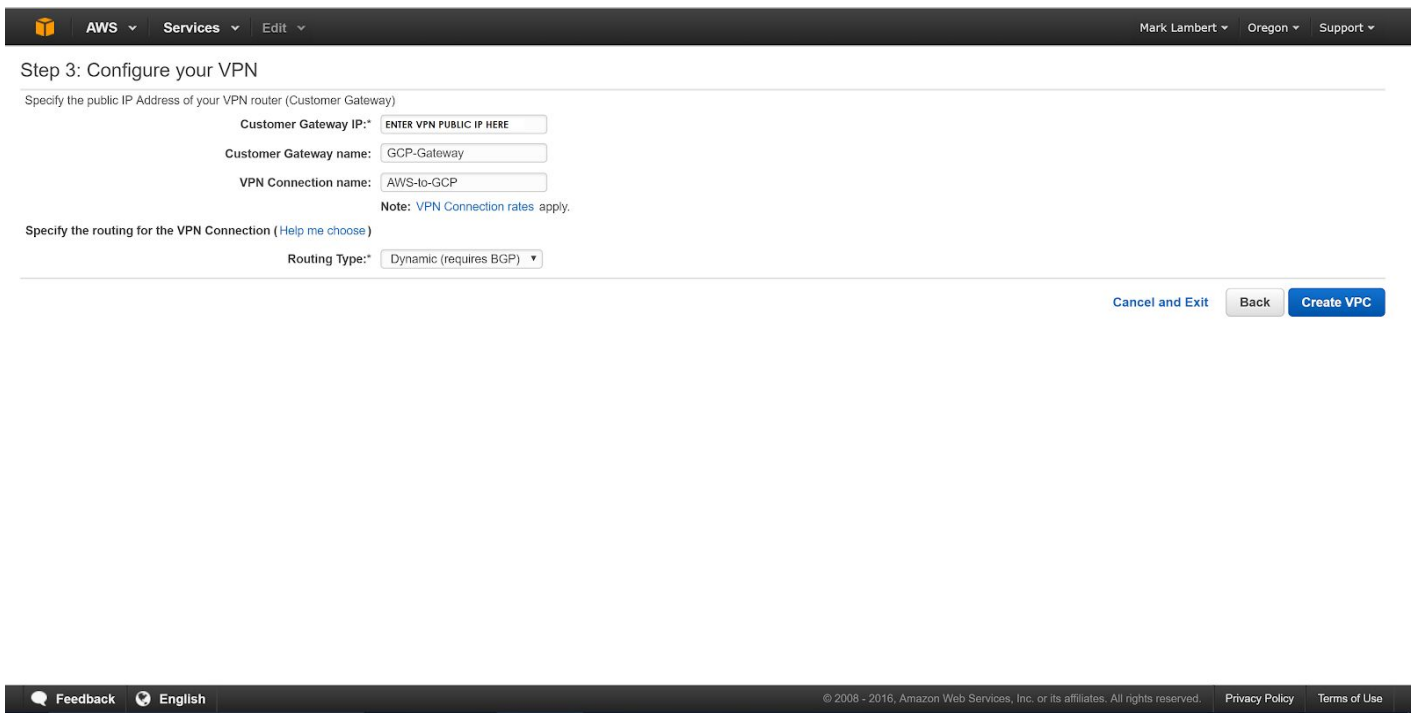
Configuring the VPN

1. To configure the VPN, enter the **Customer Gateway IP**, which is the IP address assigned to the GCP VPN gateway created in the Configuration - GCP section:

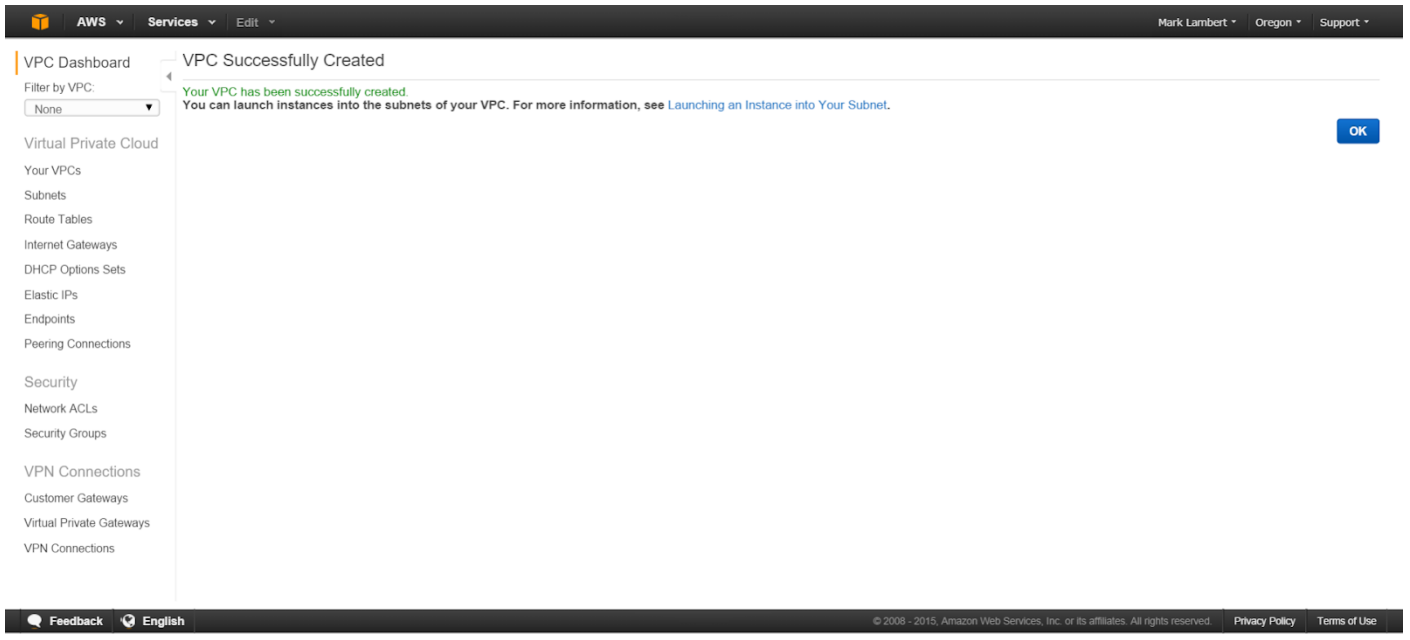


2. In addition to the Customer Gateway IP, enter a **Customer Gateway name** and a **VPN Connection name**.

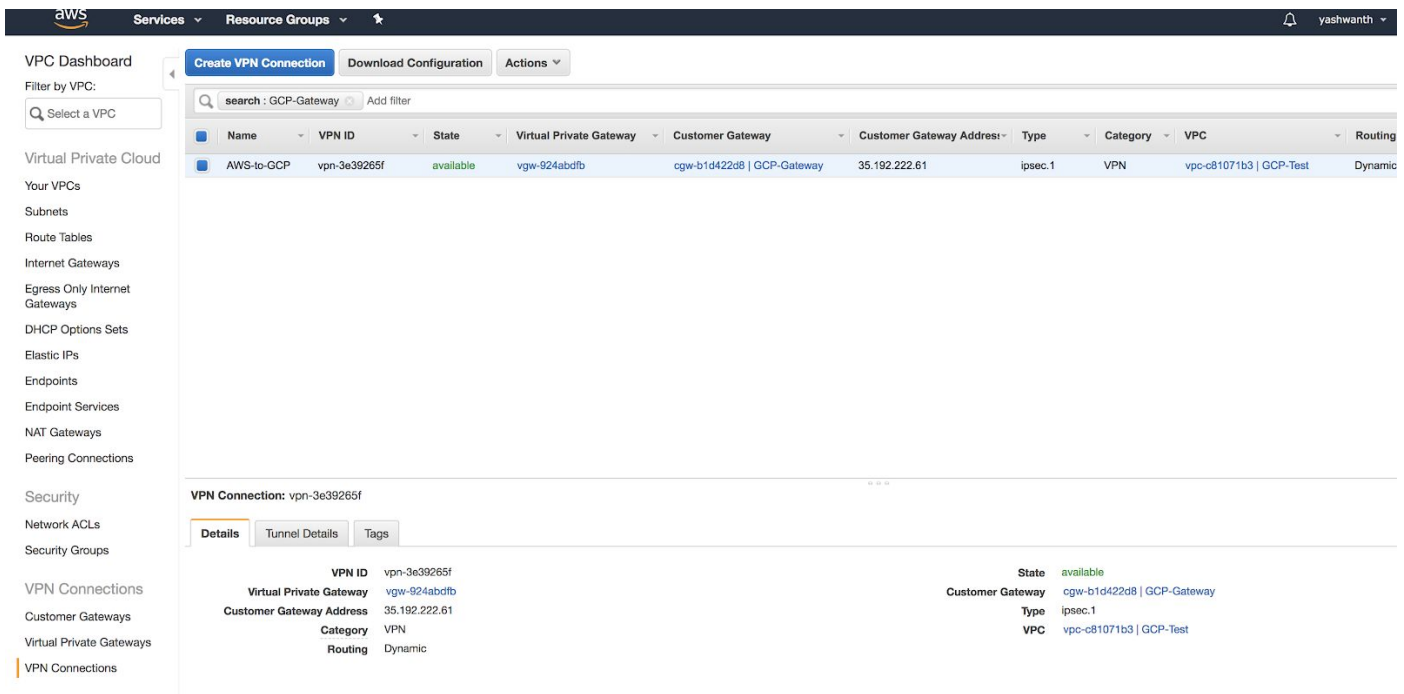
3. Choose a Routing Type for the VPN connection. This section of the guide covers VPN with BGP route management, so select **Dynamic**. Enter the GCP subnet CIDR block under **IP Prefix**, and then click **Add**:



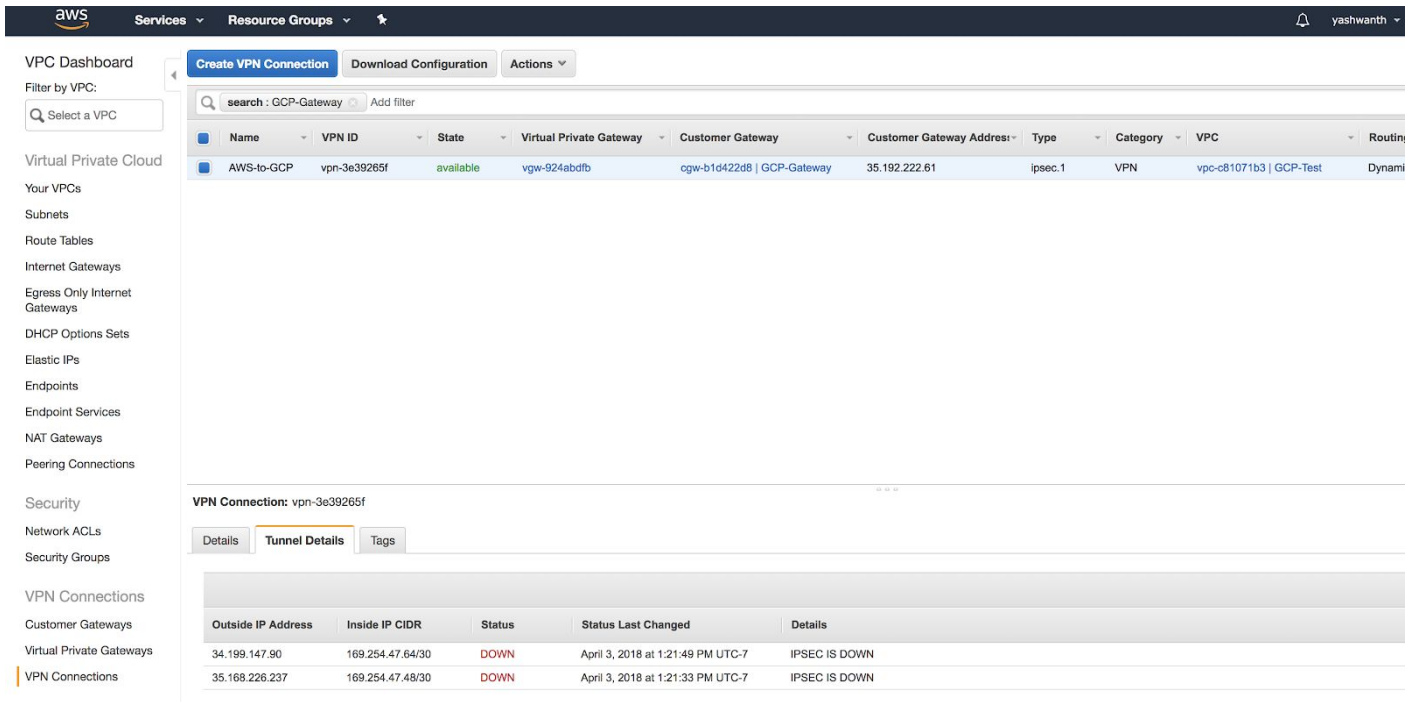
4. When you complete the required configuration, click **Create VPC** to create the new VPC network and finish the Wizard. VPC network creation takes a minute or two to complete, after which the management console status is updated:

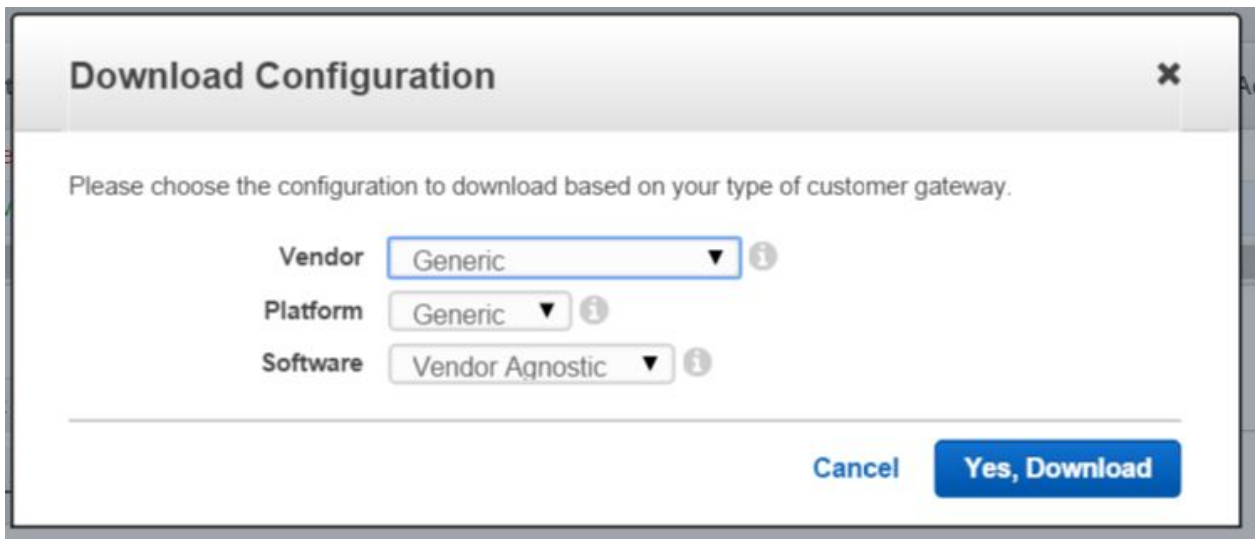


5. Select the newly-created VPC network from the Dashboard to collect the configuration detail required to complete the [GCP configuration]():



6. Because AWS requires two tunnels for redundancy, collect the IP addresses of the AWS Virtual Gateway and the pre-shared keys used for IKE authentication that are automatically generated by AWS. You can download these configuration details by clicking **Download Configuration**. Several device-specific options are available for the configuration format. For GCP, select **Generic**:





The configuration file is an ASCII text file. Note that the auto-generated pre-shared key is listed under **Pre-Shared Key** and can't be user defined. The link local address for BGP peering is listed under **Inside Addresses** and also can't be user defined.

Configuration - GCP

Google Cloud Router enables dynamic Border Gateway Protocol (BGP) route updates between your Google Cloud Platform network and your on-premises network. Cloud Router works with both legacy networks and subnets.

Configuring the VPN tunnel

1. Use the VPN creation page to create the Cloud VPN gateway and tunnels. AWS requires two tunnels for redundancy.

   Use the following parameters to configure the Cloud VPN gateway:

   - **Name:** the name of the VPN gateway.
   - **Description:** a brief description of the VPN connection.
   - **Network:** the GCP network the VPN gateway will attach to. **Note:** this is the network to which VPN connectivity will be made available.
   - **Region:** the home region of the VPN gateway. *The VPN gateway must be located in the same region as the subnets it is passing traffic through the tunnels for.* In addition, Cloud Router only programs learned routes in the region it is configured in. It will not broadcast the other routes from different regions.

- **IP address:** the static external public IP address used by the VPN gateway. Either assign an existing, unused, external static public IP address within the project or create a new one.

2. Using the following parameters, configure each tunnel managed by the Cloud VPN gateway: For tunnel 2, change the tunnel name to another unique name and the Remote peer IP address to a second on-premises IP address.

    - **Name:** the name of the tunnel.
    - **Remote peer IP address:** the public IP address of the on-premises VPN appliance which will be used to connect to Cloud VPN.
    - **IKE version:** the IKE protocol version. AWS requires **IKEv1.**
    - **Shared secret:** a shared secret used for mutual authentication by the VPN gateways. This is provided in the configuration file downloaded in the final step of the Configuration - AWS section of this document.
    - **Routing options:** Cloud VPN supports multiple routing options for the exchange of route information between the VPN gateways. For this example, use **Dynamic (BGP)** routing.. Static Routes were covered earlier in this guide.
    - **Cloud Router:** the Cloud Router instance associated with this VPN tunnel created in the Cloud Router section.
    - **BGP session:** the BGP configuration to be used by the Cloud Router for this VPN tunnel.

Google Cloud Platform VPN-testing

Create a VPN connection

A virtual private network lets you securely connect your Google Compute Engine resources to your own private network. Google VPN uses IKEv1 or IKEv2 to establish the IPSec connectivity. Learn more

Google Compute Engine VPN gateway

Name
gcp-to-aws

Description (Optional)
Google Cloud VPN to AWS VPC VPN using BGP

Network
default

Region
us-central1

IP address
gcp-to-aws-staticip (35.192.222.61)

Tunnels
You can have multiple tunnels to a single Peer VPN gateway

Name
gcp-to-aws-tunnel

Description (Optional)
GCP to AWS BGP VPN Tunnel

Remote peer IP address
34.199.147.90

IKE version
IKEv1

Shared secret
Shared Secret Generated by AWS

Routing options
Dynamic (BGP) | Route-based | Policy-based

Cloud router

Turn on global dynamic routing for network 'default' to allow this router to dynamically learn routes to and from all GCP regions on a network. If you're using an internal load balancer with VPN or Interconnect, learn how global dynamic routing may affect you .

BGP session
None

+ Add tunnel

Create Cancel

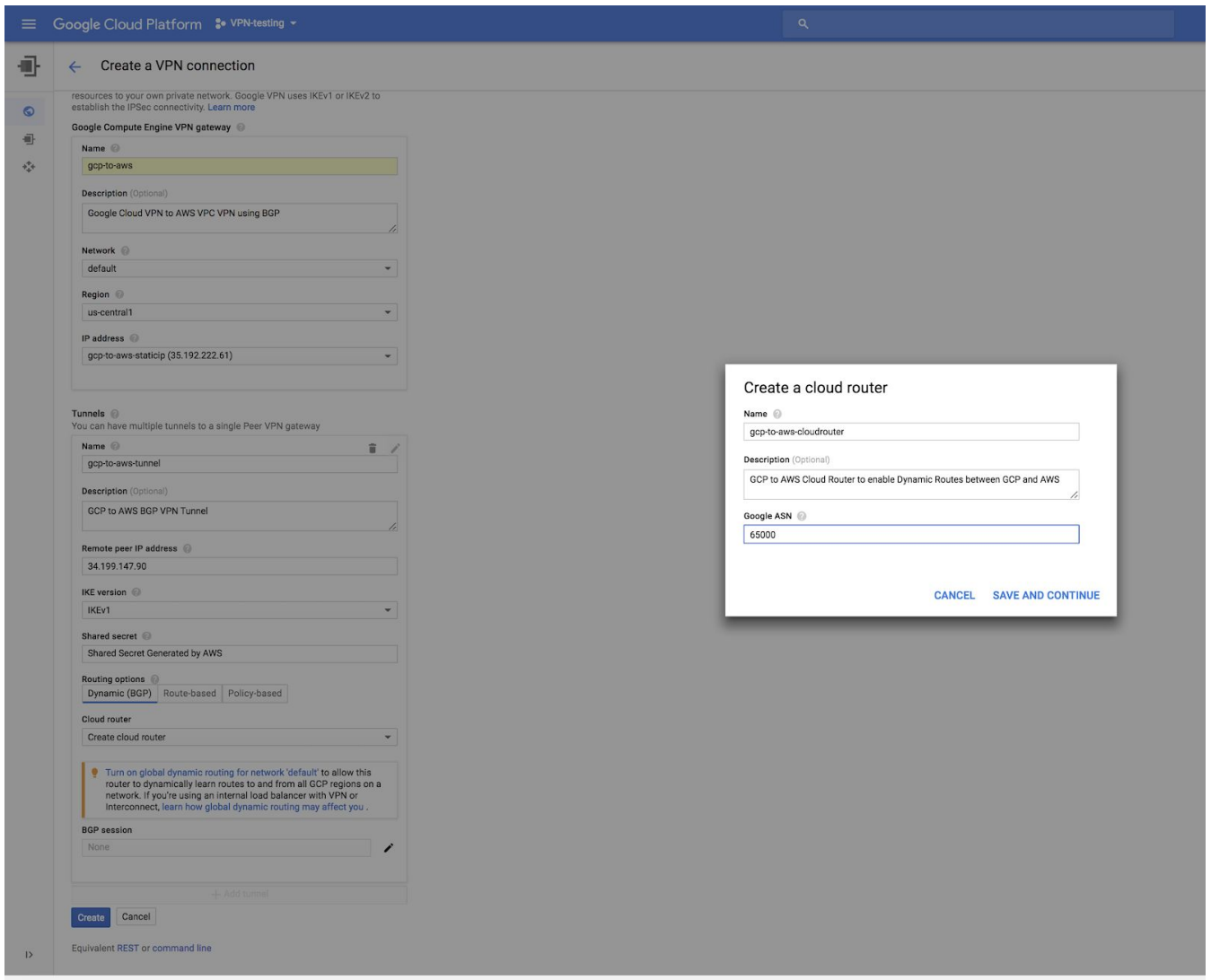
Equivalent REST or command line

Configuring the Cloud Router

1. Configure the Google Cloud Platform for site-to-site VPN connectivity using dynamic BGP is to create a new Cloud Router.

2. Click the Cloud Router dropdown menu in the VPN configuration screen, which gives you an option to create a new Cloud Router.

   Enter the all of the following required parameters:



   - **Name:** The name of the Cloud Router.
   - **Description:** A brief description of the Cloud Router.
   - **Google ASN**: The BGP Autonomous System Number (ASN) assigned to the Cloud Router. Use the ASN assigned by the Amazon VPC Creation Wizard from the configuration file downloaded in the final step of the Configuration - AWS section of this document:

```
BGP Configuration Options:
  - Customer Gateway ASN            : 65000
  - Virtual Private  Gateway ASN    : 7224
  - Neighbor IP Address             : 169.254.45.245
  - Neighbor Hold Time                     : 30
```

3. Click the Pencil icon to create a BGP connection.

4. Configure the BGP session, using the following required parameters:

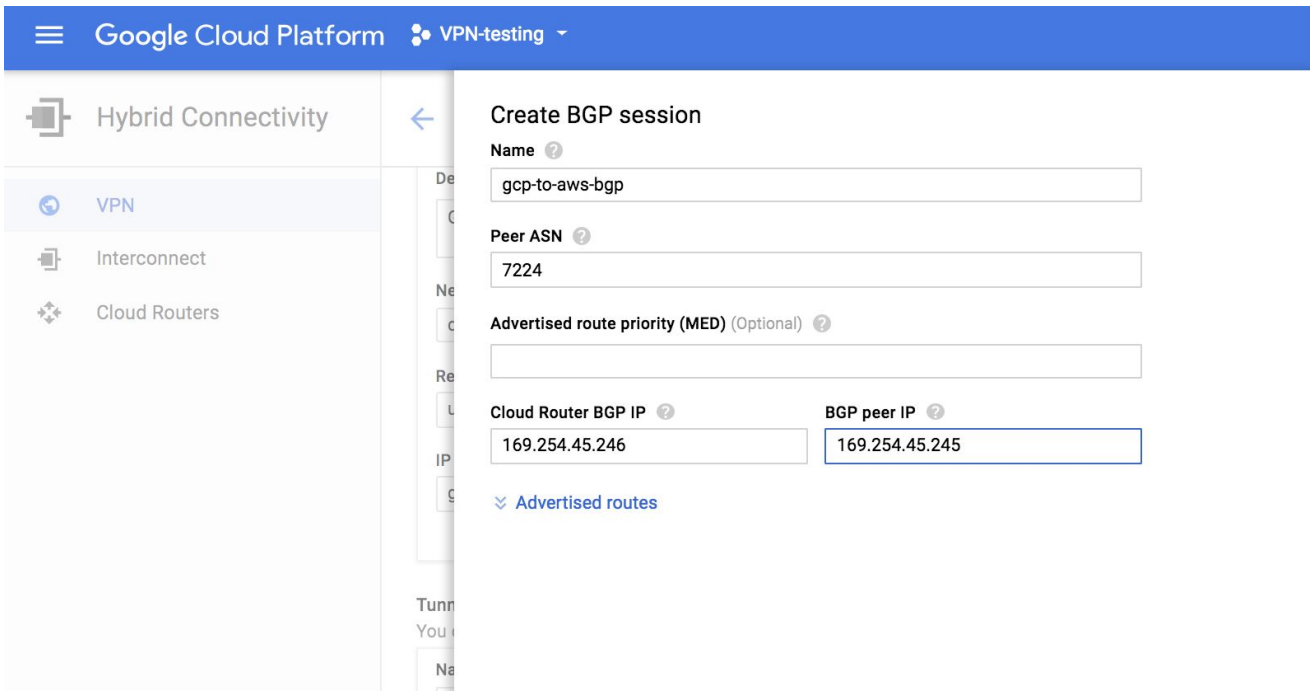
   - **Name:** The name of the BGP session
   - **Peer ASN:** Provided in the configuration file downloaded in the final step of the Configuration - AWS section of this document as the “Virtual Private Gateway ASN”:

```
BGP Configuration Options:
```

```
  - Customer Gateway ASN            : 65000
  - Virtual Private  Gateway ASN    : 7224
  - Neighbor IP Address             : 169.254.45.245
  - Neighbor Hold Time                     : 30
```

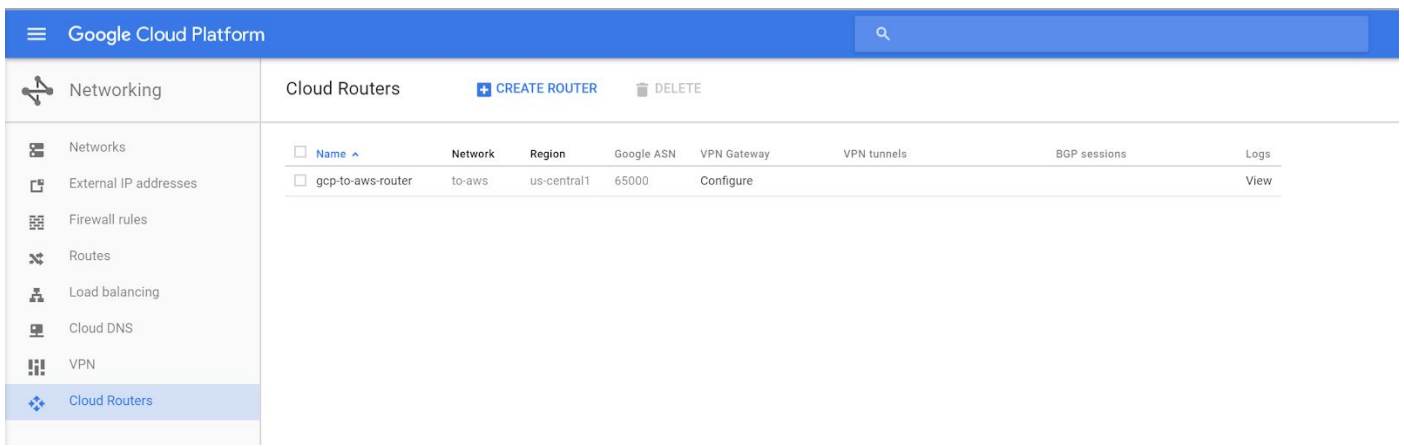
- **Google BGP IP address, Peer BGP IP address:** Provided in the configuration file downloaded in the final step of the Configuration - AWS section of this document. Note that BGP peers on a set of `169.254.x.x` link local addresses specified by the AWS configuration. “Customer Gateway” refers to the GCP side.

```
Inside IP Addresses
  - Customer Gateway                : 169.254.45.246/30
  - Virtual Private Gateway         : 169.254.45.245/30
```



5. When you’ve entered all of the BGP session info, click **Save and continue** to complete the configuration.

6. When you’ve successfully entered all information for the tunnels, on the **Create a VPN connection** form, click **Create** to create the new dual tunnel VPN connection.



Configuration - Google Cloud Router

Cloud VPN can also be configured using the gcloud command-line tool. Command-line configuration requires multiple steps.

1. Reserve a static IP address in the GCP network and region where the VPN gateway was created. Make a note of the created address for use in future steps.

```
gcloud compute addresses create vpn-static-ip --project my-project \
    --region my-region
```

2. Create the VPN gateway. Make note of the chosen name (`my-gateway`), network, and region for use in future steps:

```
gcloud compute target-vpn-gateways create my-gateway \
    --project my-project --network my-network --region my-region
```

3. Create the Cloud Router. The Amazon VPC Creation Wizard automatically assigns a BGP ASN (65000) to the Customer Gateway. Use this ASN for **the --asn** option.

```
gcloud beta compute --project my-project routers create my-router \
    --region my-region  --network my-network \
    --asn AWS-provided-customer-gateway-asn
```

4. Create the VPN tunnels referencing the **VPN gateway** and **Cloud Router** created earlier. AWS utilizes two tunnels for redundancy. Make note of the chosen tunnel names for use in future steps.

    a. Set the **peer-address** to the AWS Virtual Private Gateway IP and the **shared-secret** to the AWS assigned pre-shared key, both provided in the configuration file downloaded in the final step of the Configuration - AWS section of this document. For the second tunnel, use a unique tunnel name and change `peer-address` to the external IP address of the second AWS gateway.

    ```
    gcloud compute --project my-project vpn-tunnels create my-tunnel /
        --region my-region --ike-version 1 --target-vpn-gateway my-gateway /
        --peer-address my-AWS-virtual-private-gateway-IP /
        --shared-secret my-AWS-provided-PSK --router my-router
    ```

    b. Add the BGP link local interface. Update the Cloud Router configuration created earlier by adding a virtual interface (`--interface-name`) for the BGP peer referenced in the VPN tunnel created above. The BGP interface IP address must be the link-local IP address provided by Amazon as the **Customer Gateway Inside IP** in the configuration file downloaded in the final step of the Configuration - AWS section of this document.

    ```
    gcloud compute --project my-project routers add-interface my-router /
        --interface-name my-if /
        --ip-address my-AWS-provided-Customer-Gateway-inside-IP /
        --mask-length 30 --vpn-tunnel my-tunnel --region my-region
    ```

    c. Repeat this command for the second VPN tunnel.

5. Add the BGP peering session.

    a. Update the Cloud Router configuration by adding the BGP peer to the interface. Use the ASN and peer IP address provided by Amazon as the **Virtual Private Gateway ASN** and the **Virtual Private Gateway Inside IP** in the configuration file downloaded in the final step of the Configuration - AWS section of this document.

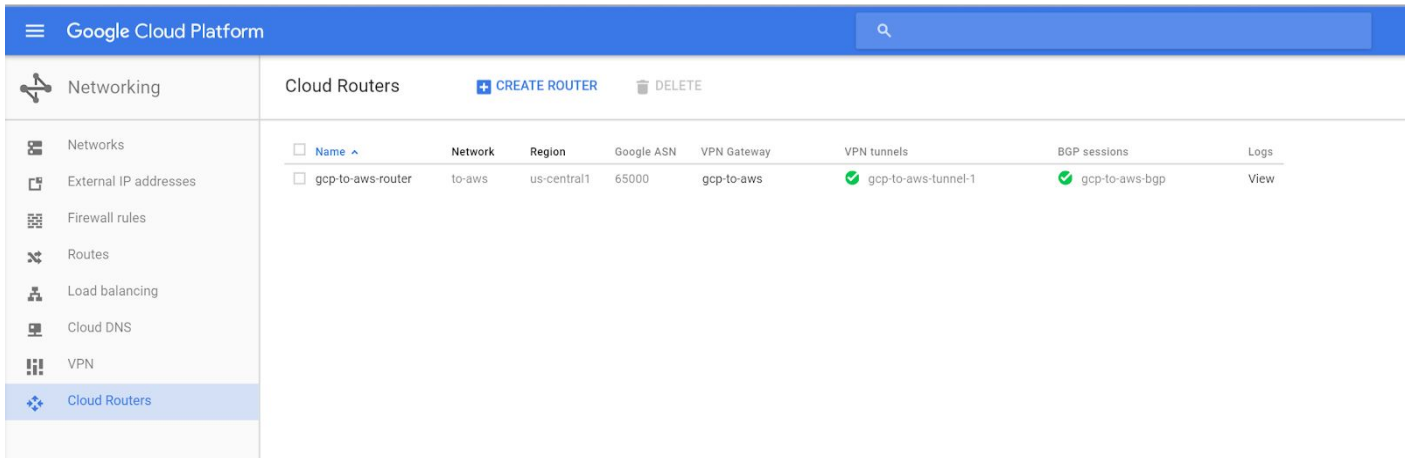
    ```
    gcloud compute --project my-project routers add-bgp-peer my-router /
        --peer-name bgp-peer1 --interface my-if /
        --peer-ip-address AWS-provided-virtual-private-gateway-inside-IP
        --peer-asn AWS-provided-virtual-private-gateway-ASN /
        --region my-region
    ```

    b. Repeat this command for the second VPN tunnel.

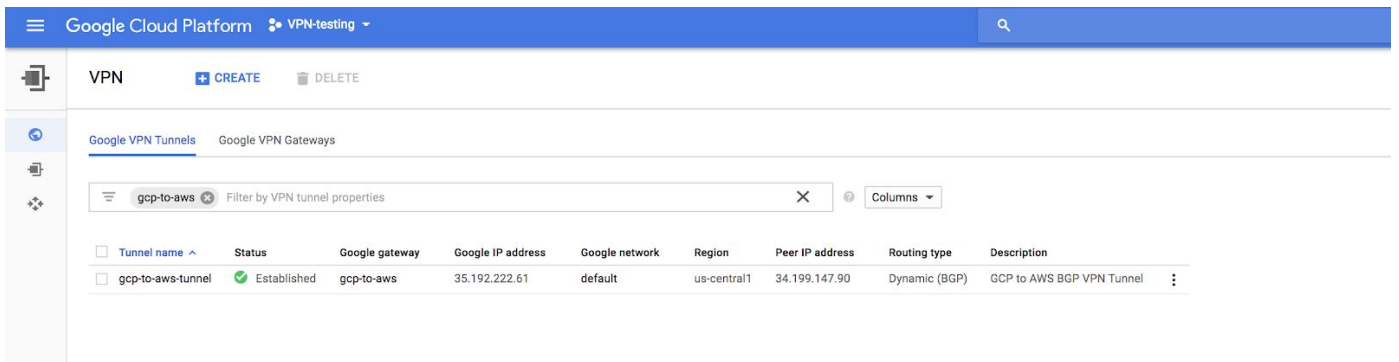
Testing the site-to-site VPN

Verifying connectivity

1. Verify that Cloud Router has successfully initiated BGP peering with AWS. Check the Cloud Router status in the GCP console for a green checkbox icon.

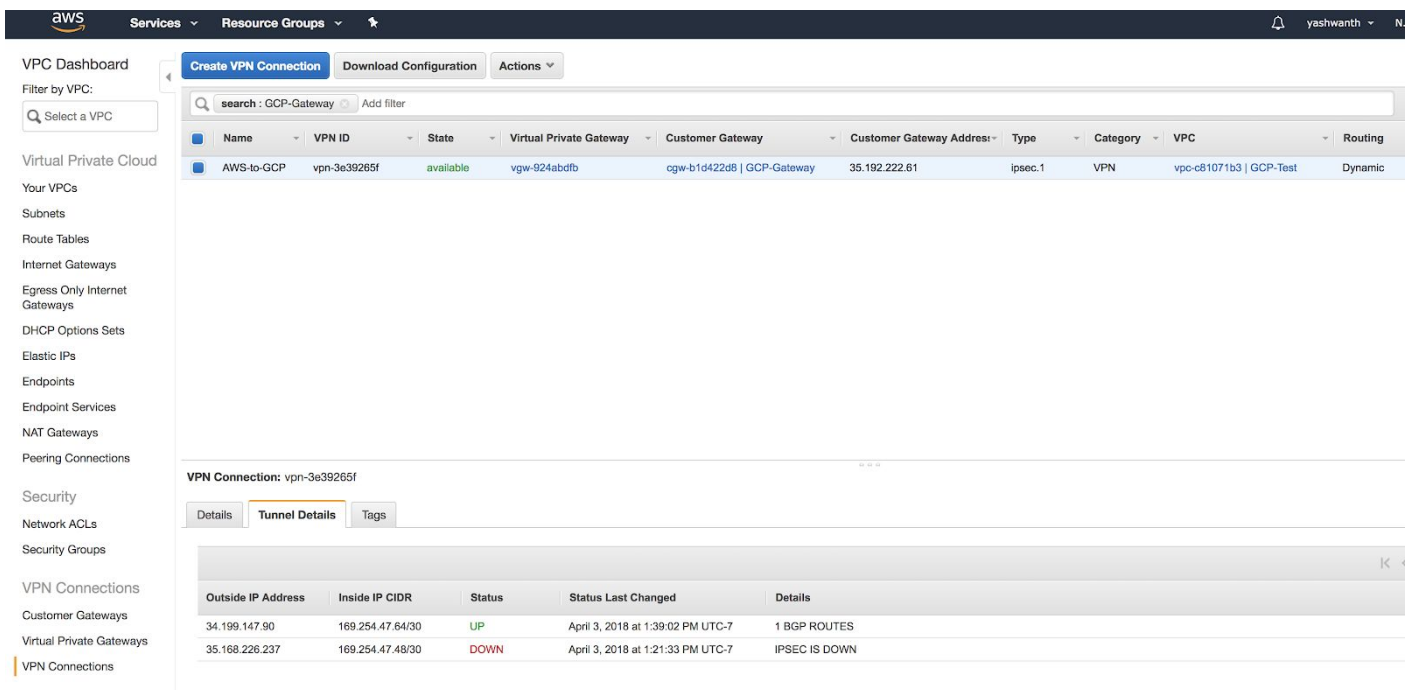


2. Verify that the IPsec tunnel has been successfully initiated. Check the VPN status in the console:



Tunnels between GCP and AWS can take a couple of minutes to establish.

3. On the AWS side, verify that the configured tunnel is up:



Note that the unconfigured tunnel will remain down unless a second tunnel was configured on the GCP side. This is expected.

Testing the VPN tunnel

With the site-to-site VPN online, the tunnel is now ready for testing.

1. First create virtual machines (VMs) in both Amazon EC2 and Google Compute Engine. Make sure to configure the VMs on a subnet that will pass traffic through the VPN tunnel.

    - See these instructions for creating Amazon EC2 virtual machines
    - See these instructions for creating virtual machines in Google Compute Engine.

1. When you've deployed virtual machines on both platforms, do an ICMP echo (ping) test to help ensure network connectivity. Note that on AWS, **Security Groups** provide firewall capabilities for EC2 instances. The default security group for a new instance does not allow ICMP. For this test to work, you must add a security group rule for ICMP.

   On the GCP side, connect using SSH into a virtual machine (VM) instance and test the connection to another machine behind the on-premises gateway.

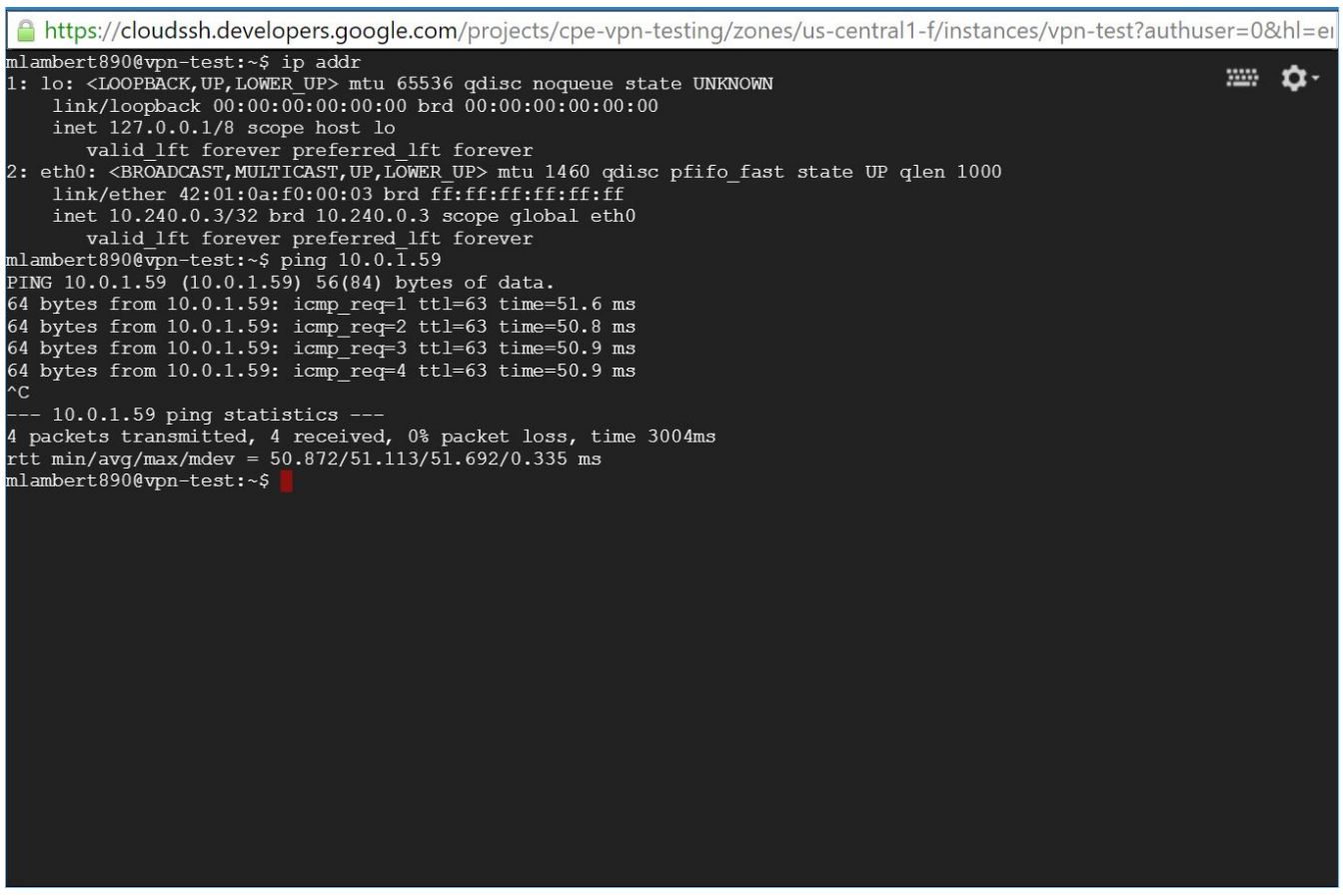
   a. In the GCP Console, from [Compute Engine, VM Instances tab](), find the GCP virtual machine you created.
   b. In the **Connect** column, click **SSH**.

      A browser window opens at the VM's command line.

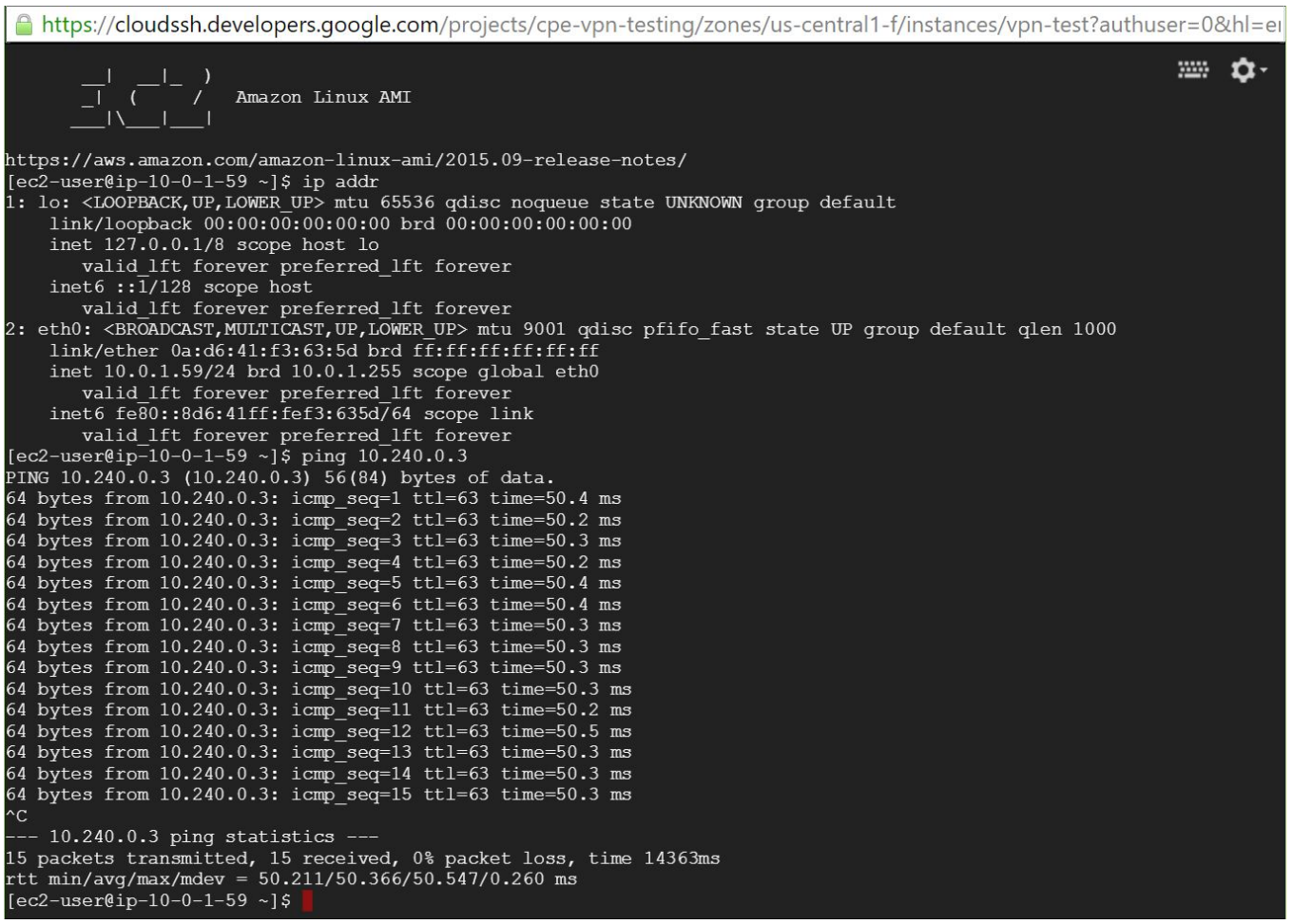
   c. Ping a machine behind the on-premises gateway to test connectivity through the VPN tunnel from the GCP side.

A demonstration of a functional tunnel follows.

A Google Compute Engine virtual machine pinging the virtual machine in Amazon EC2:



Amazon EC2 virtual machine pinging the virtual machine in Compute Engine:



Hint: The local certification used to connect by SSH into the Amazon EC2 instance must have restrictive access rights (`chmod 400 certfile`).

Troubleshooting

Suggested troubleshooting steps:

1. Verify that tunnels are shown as up on both GCP and AWS. If one or both are not, revisit the configuration steps to make sure the parameters in your GCP and AWS configurations are matching and mirroring. Make sure parameters are taken from the correct section in the downloaded AWS configuration file.
2. Verify that routes are correct on the AWS side by checking entries in the route table associated with the VPC in question. Make sure the CIDR range for GCP is routed towards the AWS VPN gateway “vgw-xxxxxx”.
3. Verify that routes are correct on the GCP side by checking the routes configured for the network in question. Traffic destined to AWS CIDR ranges should be routed towards the VPN gateway.
4. Verify that ICMP/traffic is not being blocked by checking the security group settings on the AWS side and the firewall rules on the GCP side.
5. Useful debugging tool: You can run `tcpdump` on the compute instance that is the ping target to find out which direction is not working for the ping. Here’s a sample command to capture ping packets: `sudo tcpdump -i eth0 icmp`.
6. Also see these additional troubleshooting steps.