# **Dynamic VPN gateways with Cloud Routers**

Objectives

In this lab, you learn how to perform the following tasks:

* Create two networks to represent your VPC and on-premises network, with a VM in each network to test connectivity
* Connect the two networks using Cloud Routers
* Add a new subnetwork and verify that Dynamic BGP routing is working

**Task 1. Create the networks**

Create two custom networks with subnetworks, and start micro 1vCPU VMs in each.

**Create the first network**

1. In the Cloud Console, click **Activate Cloud Shell** (Cloud Shell).
2. If prompted, click **Continue**.
3. Copy the following command in the cloud shell to create gcp-vpc network:  
     
   cloud compute networks create gcp-vpc --description=<write optional description> --subnet-mode=custom --bgp-routing-mode=regional
4. Copy the following command in the cloud shell to create subnet-a in gcp-vpc network:  
     
   gcloud compute networks subnets create subnet-a --range=10.5.4.0/24 --network=gcp-vpc --region=us-central1

### ****Create the second network****

For your second network, you use a different region than the first network.

1. Copy the following command in the cloud shell to create on-prem network:

cloud compute networks create on-prem --description=<write optional description> --subnet-mode=custom --bgp-routing-mode=regional

1. Copy the following command in the cloud shell to create subnet-b in on-prem network:  
     
   gcloud compute networks subnets create subnet-b --range=10.1.3.0/24 --network=on-prem --region=europe-west

## Task2. Create the utility VMs

### ****Create the first instance****

The first VM is created in the same region as **gcp-vpc**.

1. Copy the following command in the cloud shell to create gcp-server VM:  
     
   gcloud beta compute instances create gcp-server --zone=us-central1-a --machine-type=n1-standard-1 --subnet=subnet-a --network-tier=PREMIUM --maintenance-policy=MIGRATE --scopes=https://www.googleapis.com/auth/devstorage.read\_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append --image=debian-10-buster-v20200902 --image-project=debian-cloud --boot-disk-size=10GB --boot-disk-type=pd-standard --boot-disk-device-name=gcp-server --no-shielded-secure-boot --no-shielded-vtpm --no-shielded-integrity-monitoring --reservation-affinity=any

### ****Create the second instance****

The second VM is created in the same region as **on-prem**.

1. Copy the following command in the cloud shell to create gcp-server VM:  
     
   gcloud beta compute instances create on-prem-1 --zone=europe-west1-b --machine-type=n1-standard-1 --subnet=subnet-b --network-tier=PREMIUM --maintenance-policy=MIGRATE --scopes=https://www.googleapis.com/auth/devstorage.read\_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append --image=debian-10-buster-v20200902 --image-project=debian-cloud --boot-disk-size=10GB --boot-disk-type=pd-standard --boot-disk-device-name=on-prem-1 --no-shielded-secure-boot --no-shielded-vtpm --no-shielded-integrity-monitoring --reservation-affinity=any

### ****Allow traffic to gcp-vpc****

1. Copy the following command in the cloud shell to create firewall rules that allow ssh and icmp traffic on gcp-vpc:  
     
   gcloud compute firewall-rules create allow-icmp-ssh-gcp-vpc --direction=INGRESS --priority=1000 --network=gcp-vpc --action=ALLOW --rules=tcp:22,icmp --source-ranges=0.0.0.0/0

### ****Allow traffic to on-prem****

1. Copy the following command in the cloud shell to create firewall rules that allow ssh and icmp traffic on on-prem network:  
     
   gcloud compute firewall-rules create allow-icmp-ssh-on-prem --direction=INGRESS --priority=1000 --network=on-prem --action=ALLOW --rules=tcp:22,icmp --source-ranges=0.0.0.0/0

## Task 4. Verify network connectivity

### ****Test connectivity from gcp-server to on-prem-1****

1. Write the following command:  
     
   gcloud compute instances list  
     
   this will list VMs created with some of it properties, note the external and internal ip addresses of both VMs
2. Copy the following command in the cloud shell to ssh into gcp-server VM  
     
   gcloud compute ssh --zone us-central1-a gcp-server
3. To test connectivity to on-prem-1's external IP address, run the following command, replacing on-prem-1's external IP address with the value noted earlier:  
     
   ping -c 3 <Enter on-prem-1's external IP address here>
4. To test connectivity to on-prem-1's internal IP address, run the following command, replacing on-prem-1's internal IP address with the value noted earlier:

ping -c 3 <Enter on-prem-1's internal IP address here>  
  
You should see 100% packet loss when pinging the internal IP address

1. Write “esit” to end ssh session.

Do the same for on-prem VM

## Task 5. Create the Cloud Routers

### ****Create the gcp-vpc Cloud Router****

1. To create cloud router for gcp-vpc network run the following command:  
     
   gcloud compute routers create gcp-vpc-cr --region=us-central1 --network=gcp-vpc --asn=65470

### ****Create the on-prem Cloud Router****

1. To create cloud router run the following command  
     
   gcloud compute routers create on-prem-cr --region=europe-west1 --network=on-prem --asn=65503

### ****Prepare for VPN Gateways configuration****

The Cloud Console displays rows labeled **gcp-vpc-cr** and **on-prem-cr** with their respective regions.

You need a static External IP address for each gateway. Create them now and note the IP addresses. You will use them later in the lab.

1. To reserve static IP for gcp-vpc vpn tunnel run the following command:  
     
   gcloud compute addresses create gcp-vpc-ip --region=us-central1
2. To reserve static IP for on-prem vpn tunnel run the following command:  
     
   gcloud compute addresses create on-prem-ip --region=europe-west1
3. Write the following to list reserved IP addresses:  
     
   gcloud compute addresses list  
     
   make note of the addresses as we will use it in the nest steps

### ****Create the first VPN****

1. To create first vpn gateway run the following command:  
     
   gcloud compute target-vpn-gateways create "vpn-1" --region "us-central1" --network "gcp-vpc"
2. To create forwarding rules for ESP, UDP 500 and UDP 4500, run the following commands:  
     
   gcloud compute forwarding-rules create "vpn-1-rule-esp" --region "us-central1" --address "gcp-vpc-ip" --ip-protocol "ESP" --target-vpn-gateway "vpn-1"

gcloud compute forwarding-rules create "vpn-1-rule-udp500" --region "us-central1" --address "gcp-vpc-ip" --ip-protocol "UDP" --ports "500" --target-vpn-gateway "vpn-1"

gcloud compute forwarding-rules create "vpn-1-rule-udp4500" --region "us-central1" --address "gcp-vpc-ip" --ip-protocol "UDP" --ports "4500" --target-vpn-gateway "vpn-1"

1. To create first vpn tunnel run the following command:  
     
   gcloud compute vpn-tunnels create "vpn-1-tunnel-1" --region "us-central1" --peer-address <on-prem reserved ip> --shared-secret "gcprocks" --ike-version "2" --router "gcp-vpc-cr" --target-vpn-gateway "vpn-1"
2. To create BGP sesiion for gcp-vpc-cr router run the following commands:  
     
   gcloud compute routers add-interface gcp-vpc-cr --interface-name "bgp1to2" --vpn-tunnel "vpn-1-tunnel-1" --ip-address "169.254.0.1" --mask-length 30 --region "us-central1"  
     
   gcloud compute routers add-bgp-peer gcp-vpc-cr --peer-asn "65503" -interface "bgp1to2" peer-ip-address "169.254.0.2" --region "us-central1" --peer-name "private-session"

### ****Create the second VPN****

1. To create second vpn gateway run the following command:  
     
   gcloud compute target-vpn-gateways create "vpn-2" --region "europe-west1" --network "on-prem"
2. To create forwarding rules for ESP, UDP 500 and UDP 4500, run the following commands:  
     
   gcloud compute forwarding-rules create "vpn-2-rule-esp" --region "europe-west1" --address "on-prem-ip" --ip-protocol "ESP" --target-vpn-gateway "vpn-2"

gcloud compute forwarding-rules create "vpn-2-rule-udp500" --region "europe-west1" --address "on-prem-ip" --ip-protocol "UDP" --ports "500" --target-vpn-gateway "vpn-2"  
  
gcloud compute forwarding-rules create "vpn-2-rule-udp4500" --region "europe-west1" --address "on-prem-ip" --ip-protocol "UDP" --ports "4500" --target-vpn-gateway "vpn-2""

1. To create second vpn tunnel run the following command:  
     
   gcloud compute vpn-tunnels create "vpn-1-tunnel-2" --region "europe-west1" --peer-address "34.72.114.142" --shared-secret "gcprocks" --ike-version "2" --router "on-prem-cr" --target-vpn-gateway "vpn-2"
2. To create BGP sesiion for gcp-vpc-cr router run the following commands:  
     
   gcloud compute routers add-interface on-prem-cr --interface-name "bgp2to1" --vpn-tunnel "vpn-1-tunnel-2" --ip-address "169.254.0.2" --mask-length 30 --region "europe-west1"

gcloud compute routers add-bgp-peer gcp-vpc-cr --peer-asn "65470" -interface "bgp2to1" peer-ip-address "169.254.0.1" --region "europe-west1" --peer-name "private-session"

## Task 6. Verify connection

1. Copy the following command in the cloud shell to ssh into gcp-server VM  
     
   gcloud compute ssh --zone us-central1-a gcp-server
2. To test connectivity to on-prem-1's internal IP address, run the following command, replacing on-prem-1's internal IP address with the value noted earlier:

ping -c 3 <Enter on-prem-1's internal IP address here>

1. Write “exit” to end ssh session
2. opy the following command in the cloud shell to ssh into on-prem-1 VM  
     
   gcloud compute ssh --zone europe-west1-b on-prem-1
3. To test connectivity to gcp-server's internal IP address, run the following command, replacing gcp-server's internal IP address with the value noted earlier:

ping -c 3 <Enter gcp-server's internal IP address here>

1. Write “exit” to end ssh session

## Task 7. Demonstrate Dynamic Routing

### ****Create a new subnet in on-prem****

1. To create subnet-c in the on-prem network run the following command  
     
   gcloud compute networks subnets create subnet-c --range=10.4.2.0/24 --network=on-prem --region=europe-west1

### ****Create a new utility VM in the new subnet****

1. Run the following command to create on-prem-2 VM:  
     
   gcloud beta compute instances create on-prem-2 --zone=europe-west1-c --machine-type=n1-standard-1 --subnet=subnet-c --network-tier=PREMIUM --maintenance-policy=MIGRATE --scopes=https://www.googleapis.com/auth/devstorage.read\_only,https://www.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.googleapis.com/auth/service.management.readonly,https://www.googleapis.com/auth/trace.append --image=debian-10-buster-v20200902 --image-project=debian-cloud --boot-disk-size=10GB --boot-disk-type=pd-standard --boot-disk-device-name=on-prem-2 --no-shielded-secure-boot --no-shielded-vtpm --no-shielded-integrity-monitoring --reservation-affinity=any

### ****Test connectivity****

1. Write the following command:  
     
   gcloud compute instances list
2. Note on-prem-2 internal IP addresses
3. Copy the following command in the cloud shell to ssh into gcp-server VM  
     
   gcloud compute ssh --zone us-central1-a gcp-server
4. To test connectivity to on-prem-2's internal IP address, run the following command, replacing on-prem-2's internal IP address with the value noted earlier:

ping -c 3 <Enter on-prem-2's internal IP address here>

1. Write “exit” to end ssh session

### ****View the routes****

1. To view dynamic routes and router status run the following command:  
     
   gcloud compute routers get-status gcp-vpc-cr --region us-central1

### ****Remove the external IP addresses****

1. Run the following command to stop gcp-server VM  
     
   gcloud compute instances stop gcp-server –zone us-central1-a  
     
   Instances need to be stopped before you can make changes to their network interfaces  
   .
2. Run the following command in the cloud shell:  
     
   gcloud compute instances describe gcp-server –xone us-central1-a  
     
   under networkInterface 🡪 accessConfigs make note of the name value we will use it in the next step
3. To remove external ip address of gcp-server run the following command:  
     
   gcloud compute instances delete-access-config gcp-server --access-config-name <access config name> --zone us-central1-a
4. Run the following command to stop gcp-server VM  
     
   gcloud compute instances stop gcp-server –zone us-central1-a
5. To verify that external ip address is deleted run the following command  
     
   gcloud compute instances list  
     
   note that gcp-server has no external ip address