Demo VPN: Using Cloud Router + Cloud HA VPN + BGP

- Plan for the demo:
    - **No need for Org node here**
    - Create 2 projects - one to act as cloud project and other as on-prem project
    - Rename one project as **gcp-project** and other as **on-premise** project
    - Use **BGP peering using dynamic routing** to update the routes automatically
    - Not using any enterprise level hardwares or routers to setup actual On-Premise

========== Cloud ==========
declare -x cloud_proj_id="centered-flash-353712"
gcloud config set project $cloud_proj_id

Create VPC in gcp-project
gcloud compute networks create cloud-vpc --project=$cloud_proj_id --description=VPC\ for\ GCP\ Cloud\ side --subnet-mode=custom --mtu=1460 --bgp-routing-mode=regional

Add 2 Subnets to VPC for us-central1 and us-east1 region:
gcloud compute networks subnets create cloud-vpc-subnet1 --project=$cloud_proj_id --range=10.1.1.0/24 --stack-type=IPV4_ONLY --network=cloud-vpc --region=us-central1

gcloud compute networks subnets create cloud-vpc-subnet2 --project=$cloud_proj_id --range=10.2.1.0/24 --stack-type=IPV4_ONLY --network=cloud-vpc --region=us-east1

Add default 4 FW rules
gcloud compute firewall-rules create cloud-vpc-allow-custom --project=$cloud_proj_id --network=projects/centered-flash-353712/global/networks/cloud-vpc --description=Allows\ connection\ from\ any\ source\ to\ any\ instance\ on\

the\ network\ using\ custom\ protocols. --direction=INGRESS
--priority=65534 --source-ranges=10.1.1.0/24,10.2.1.0/24 --action=ALLOW
--rules=all

gcloud compute firewall-rules create cloud-vpc-allow-icmp
--project=$cloud_proj_id
--network=projects/centered-flash-353712/global/networks/cloud-vpc
--description=Allows\ ICMP\ connections\ from\ any\ source\ to\ any\
instance\ on\ the\ network. --direction=INGRESS --priority=65534
--source-ranges=0.0.0.0/0 --action=ALLOW --rules=icmp

gcloud compute firewall-rules create cloud-vpc-allow-rdp
--project=$cloud_proj_id
--network=projects/centered-flash-353712/global/networks/cloud-vpc
--description=Allows\ RDP\ connections\ from\ any\ source\ to\ any\
instance\ on\ the\ network\ using\ port\ 3389. --direction=INGRESS
--priority=65534 --source-ranges=0.0.0.0/0 --action=ALLOW
--rules=tcp:3389

gcloud compute firewall-rules create cloud-vpc-allow-ssh
--project=$cloud_proj_id
--network=projects/centered-flash-353712/global/networks/cloud-vpc
--description=Allows\ TCP\ connections\ from\ any\ source\ to\ any\
instance\ on\ the\ network\ using\ port\ 22. --direction=INGRESS
--priority=65534 --source-ranges=0.0.0.0/0 --action=ALLOW --rules=tcp:22

========== On Premise ==========
declare -x onprem_proj_id="gcp-cloud-service"
gcloud config set project $onprem_proj_id

Create VPC in on-premise
gcloud compute networks create onprem-vpc --project=$onprem_proj_id
--description=VPC\ for\ OnPremise\ side --subnet-mode=custom
--mtu=1460 --bgp-routing-mode=regional

Add 1 Subnet to VPC for us-central1 region:

```
gcloud compute networks subnets create onprem-vpc-subnet1
--project=$onprem_proj_id --range=192.168.1.0/24
--stack-type=IPV4_ONLY --network=onprem-vpc --region=us-central1
```

Add default 4 FW rules

```
gcloud compute firewall-rules create onprem-vpc-allow-custom
--project=$onprem_proj_id
--network=projects/gcp-cloud-service/global/networks/onprem-vpc
--description=Allows\ connection\ from\ any\ source\ to\ any\ instance\ on\
the\ network\ using\ custom\ protocols. --direction=INGRESS
--priority=65534 --source-ranges=192.168.1.0/24 --action=ALLOW
--rules=all

gcloud compute firewall-rules create onprem-vpc-allow-icmp
--project=$onprem_proj_id
--network=projects/gcp-cloud-service/global/networks/onprem-vpc
--description=Allows\ ICMP\ connections\ from\ any\ source\ to\ any\
instance\ on\ the\ network. --direction=INGRESS --priority=65534
--source-ranges=0.0.0.0/0 --action=ALLOW --rules=icmp

gcloud compute firewall-rules create onprem-vpc-allow-rdp
--project=$onprem_proj_id
--network=projects/gcp-cloud-service/global/networks/onprem-vpc
--description=Allows\ RDP\ connections\ from\ any\ source\ to\ any\
instance\ on\ the\ network\ using\ port\ 3389. --direction=INGRESS
--priority=65534 --source-ranges=0.0.0.0/0 --action=ALLOW
--rules=tcp:3389

gcloud compute firewall-rules create onprem-vpc-allow-ssh
--project=$onprem_proj_id
--network=projects/gcp-cloud-service/global/networks/onprem-vpc
--description=Allows\ TCP\ connections\ from\ any\ source\ to\ any\
instance\ on\ the\ network\ using\ port\ 22. --direction=INGRESS
--priority=65534 --source-ranges=0.0.0.0/0 --action=ALLOW --rules=tcp:22
```

========== Cloud ==========

Create a VM in cloud-vpc

```
gcloud compute instances create cloud-vpc-instance1
--project=$cloud_proj_id --zone=us-central1-b --machine-type=f1-micro
--network-interface=subnet=cloud-vpc-subnet1,no-address
--maintenance-policy=MIGRATE --provisioning-model=STANDARD
--scopes=https://www.googleapis.com/auth/devstorage.read_only,https://w
ww.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/mo
nitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.g
oogleapis.com/auth/service.management.readonly,https://www.googleapis.
com/auth/trace.append
--create-disk=auto-delete=yes,boot=yes,device-name=cloud-vpc-instance1,
image=projects/debian-cloud/global/images/debian-11-bullseye-v20220621
,mode=rw,size=10,type=projects/$cloud_proj_id/zones/us-central1-a/diskTy
pes/pd-balanced --no-shielded-secure-boot --shielded-vtpm
--shielded-integrity-monitoring --reservation-affinity=any
```

========== On Premise ==========

Create a VM in onprem-vpc

```
gcloud compute instances create onprem-vpc-instance1
--project=$onprem_proj_id --zone=us-central1-a --machine-type=f1-micro
--network-interface=subnet=onprem-vpc-subnet1,no-address
--maintenance-policy=MIGRATE --provisioning-model=STANDARD
--scopes=https://www.googleapis.com/auth/devstorage.read_only,https://w
ww.googleapis.com/auth/logging.write,https://www.googleapis.com/auth/mo
nitoring.write,https://www.googleapis.com/auth/servicecontrol,https://www.g
oogleapis.com/auth/service.management.readonly,https://www.googleapis.
com/auth/trace.append
--create-disk=auto-delete=yes,boot=yes,device-name=cloud-vpc-instance1,
image=projects/debian-cloud/global/images/debian-11-bullseye-v20220621
,mode=rw,size=10,type=projects/$onprem_proj_id/zones/us-central1-a/disk
Types/pd-balanced --no-shielded-secure-boot --shielded-vtpm
--shielded-integrity-monitoring --reservation-affinity=any
```

Connectivity Check using Private IP
SSH into cloud-vpc-instance1 and ping private ip of the onprem-vpc-instance1 - It should not work
Vice-versa - It should not work

========== Cloud HA VPN Set Up ==========
Create a Cloud HA-VPN Gateway in network cloud-vpc:
gcloud compute vpn-gateways create cloud-vpc-vpn-gw1 --network cloud-vpc --region us-central1 --project=$cloud_proj_id

Create a Cloud HA-VPN Gateway in network onprem-vpc:
gcloud compute vpn-gateways create onprem-vpc-vpn-gw1 --network onprem-vpc --region us-central1 --project=$onprem_proj_id

===== Optional ====
View details of vpn-gateway cloud-vpc-vpn-gw1:
gcloud compute vpn-gateways describe cloud-vpc-vpn-gw1 --region us-central1 --project=$cloud_proj_id

View details of vpn-gateway onprem-vpc-vpn-gw1:
gcloud compute vpn-gateways describe onprem-vpc-vpn-gw1 --region us-central1 --project=$onprem_proj_id
==========

Create a cloud router in network cloud-vpc:
gcloud compute routers create cloud-vpc-router1 --region us-central1 --network cloud-vpc --asn 65001 --project=$cloud_proj_id

Create a cloud router in network onprem-vpc:
gcloud compute routers create onprem-vpc-router1 --region us-central1 --network onprem-vpc --asn 65002 --project=$onprem_proj_id

========== Cloud ==========
IKE: Internet Key Exchange protocol

- ensures the security of the VPN over public internet
- Two versions. V1 and V2
- V2 -
    - is more advanced
    - supports cryptographic mechanisms for packet transfer
    - Less bandwidth requirement
    - Supported by variety of devices like smartphones etc
    - Robust against attacks etc

Routing options:

- Dynamic BGP - Update the routes automatically whenever new subnets are added to any network. It is being deprecated now.
- Route Based - For route specific like we can give specific CIDR range for routes to be exchanged. New rack or subnets will not be updated if it is outside the CIDR range given.
- Policy Based - Policy based is also similar. Here you can provide remote and local IP address range as well. Only some specific network should be exchanged.

Create the first VPN tunnels in network cloud-vpc:

```
gcloud compute vpn-tunnels create cloud-vpc-tunnel0 \
    --peer-gcp-gateway
projects/$onprem_proj_id/regions/us-central1/vpnGateways/onprem-vpc-vpn-gw1 \
    --region us-central1 \
    --ike-version 2 \
    --shared-secret 1234 \
    --router cloud-vpc-router1 \
    --vpn-gateway cloud-vpc-vpn-gw1 \
    --interface 0 --project $cloud_proj_id
```

Create the second VPN tunnels in network cloud-vpc:

```
gcloud compute vpn-tunnels create cloud-vpc-tunnel1 \
```

```
    --peer-gcp-gateway
projects/$onprem_proj_id/regions/us-central1/vpnGateways/onprem-vpc-vpn-gw1 \
    --region us-central1 \
    --ike-version 2 \
    --shared-secret 1234 \
    --router cloud-vpc-router1 \
    --vpn-gateway cloud-vpc-vpn-gw1 \
    --interface 1 --project $cloud_proj_id
```

========== On Premise ==========

Create two vpn tunnels in network onprem-vpc

First tunnel:

```
gcloud compute vpn-tunnels create onprem-vpc-tunnel0 \
    --peer-gcp-gateway
projects/$cloud_proj_id/regions/us-central1/vpnGateways/cloud-vpc-vpn-gw1 \
    --region us-central1 \
    --ike-version 2 \
    --shared-secret 1234 \
    --router onprem-vpc-router1 \
    --vpn-gateway onprem-vpc-vpn-gw1 \
    --interface 0 --project $onprem_proj_id
```

Second Tunnel:

```
gcloud compute vpn-tunnels create onprem-vpc-tunnel1 \
    --peer-gcp-gateway
projects/$cloud_proj_id/regions/us-central1/vpnGateways/cloud-vpc-vpn-gw1 \
    --region us-central1 \
    --ike-version 2 \
    --shared-secret 1234 \
    --router onprem-vpc-router1 \
    --vpn-gateway onprem-vpc-vpn-gw1 \
```

```
    --interface 1 --project $onprem_proj_id
```

========== Cloud ==========

ASN: Autonomous System Number -

- It is like role number in school
- We can assign these numbers to our routers and then we can use them to identify against each other

Create BGP Sessions: First

Create the router interface for tunnel0 in network cloud-vpc:

```
gcloud compute routers add-interface cloud-vpc-router1 \
    --interface-name if-tunnel0-to-on-prem \
    --ip-address 169.254.0.1 \
    --mask-length 30 \
    --vpn-tunnel cloud-vpc-tunnel0 \
    --region us-central1 --project $cloud_proj_id
```

And the bgp peer for tunnel0 in network cloud-vpc:

```
gcloud compute routers add-bgp-peer cloud-vpc-router1 \
    --peer-name bgp-on-prem-tunnel0 \
    --interface if-tunnel0-to-on-prem \
    --peer-ip-address 169.254.0.2 \
    --peer-asn 65002 \
    --region us-central1 --project $cloud_proj_id
```

Create BGP Sessions: Second

Create router interface for tunnel1 in network cloud-vpc:

```
gcloud compute routers add-interface cloud-vpc-router1 \
    --interface-name if-tunnel1-to-on-prem \
    --ip-address 169.254.1.1 \
    --mask-length 30 \
    --vpn-tunnel cloud-vpc-tunnel1 \
    --region us-central1 --project $cloud_proj_id
```

And the bgp peer for tunnel1 in network cloud-vpc:

```
gcloud compute routers add-bgp-peer cloud-vpc-router1 \
        --peer-name bgp-on-prem-tunnel1 \
        --interface if-tunnel1-to-on-prem \
        --peer-ip-address 169.254.1.2 \
        --peer-asn 65002 \
        --region us-central1 --project $cloud_proj_id
```

========== On Premise ==========

Create BGP Sessions: First

Create the router interface for tunnel0 in network onprem-vpc:

```
gcloud compute routers add-interface onprem-vpc-router1 \
        --interface-name if-tunnel0-to-cloud-vpc \
        --ip-address 169.254.0.2 \
        --mask-length 30 \
        --vpn-tunnel onprem-vpc-tunnel0 \
        --region us-central1 --project $onprem_proj_id
```

And the bgp peer for tunnel0 in network onprem-vpc:

```
gcloud compute routers add-bgp-peer onprem-vpc-router1 \
        --peer-name bgp-cloud-vpc-tunnel0 \
        --interface if-tunnel0-to-cloud-vpc \
        --peer-ip-address 169.254.0.1 \
        --peer-asn 65001 \
        --region us-central1 --project $onprem_proj_id
```

Create BGP Sessions: Second

Create the router interface for tunnel1 in network onprem-vpc:

```
gcloud compute routers add-interface onprem-vpc-router1 \
        --interface-name if-tunnel1-to-cloud-vpc \
        --ip-address 169.254.1.2 \
        --mask-length 30 \
        --vpn-tunnel onprem-vpc-tunnel1 \
        --region us-central1 --project $onprem_proj_id
```

And the bgp peer for tunnel0 in network onprem-vpc:

```
gcloud compute routers add-bgp-peer onprem-vpc-router1 \
        --peer-name bgp-cloud-vpc-tunnel1 \
        --interface if-tunnel1-to-cloud-vpc \
        --peer-ip-address 169.254.1.1 \
        --peer-asn 65001 \
        --region us-central1 --project $onprem_proj_id
```

========== Cloud ==========

Verify cloud router configurations:

```
gcloud compute routers describe cloud-vpc-router1 --region us-central1
--project $cloud_proj_id
```

Reference if any issue:

========== On Premise ==========

Verify onprem router configurations:

```
gcloud compute routers describe onprem-vpc-router1 --region us-central1
--project $onprem_proj_id
```

Reference if any issue:

Release / Delete all resources

TUNNEL -> ROUTER -> GATEWAYS -> VMs -> FW -> SUBNETS -> VPC

```
gcloud compute vpn-tunnels delete onprem-vpc-tunnel0  --region
us-central1 --project $onprem_proj_id
gcloud compute vpn-tunnels delete cloud-vpc-tunnel1  --region us-central1
--project $cloud_proj_id
gcloud compute vpn-tunnels delete onprem-vpc-tunnel1 --region
us-central1 --project $onprem_proj_id
```

```
gcloud compute  routers delete cloud-vpc-router1 --region us-central1
--project $cloud_proj_id
gcloud compute  routers delete onprem-vpc-router1 --region us-central1
--project $onprem_proj_id


gcloud compute vpn-gateways delete cloud-vpc-vpn-gw1 --region
us-central1 --project $cloud_proj_id
gcloud beta compute vpn-gateways delete onprem-vpc-vpn-gw1 --region
us-central1 --project $onprem_proj_id

gcloud compute instances delete cloud-vpc-instance1 --zone us-central1-b
--project $cloud_proj_id
gcloud compute instances delete onprem-vpc-instance1 --zone
us-central1-a --project $onprem_proj_id

gcloud compute firewall-rules delete cloud-vpc-allow-custom
--project=$cloud_proj_id -q
gcloud compute firewall-rules delete cloud-vpc-allow-icmp
--project=$cloud_proj_id -q
gcloud compute firewall-rules delete cloud-vpc-allow-rdp
--project=$cloud_proj_id -q
gcloud compute firewall-rules delete cloud-vpc-allow-ssh
--project=$cloud_proj_id -q

gcloud compute firewall-rules delete onprem-vpc-allow-custom
--project=$onprem_proj_id -q
gcloud compute firewall-rules delete onprem-vpc-allow-icmp
--project=$onprem_proj_id -q
gcloud compute firewall-rules delete onprem-vpc-allow-rdp
--project=$onprem_proj_id -q
gcloud compute firewall-rules delete onprem-vpc-allow-ssh
--project=$onprem_proj_id -q
```

```
gcloud compute networks subnets delete cloud-vpc-subnet1
--region=us-central1 --project=$cloud_proj_id -q
gcloud compute networks subnets delete cloud-vpc-subnet2
--region=us-east1 --project=$cloud_proj_id -q

gcloud compute networks subnets delete onprem-vpc-subnet1
--region=us-central1 --project=$onprem_proj_id -q
gcloud compute networks subnets delete onprem-vpc-subnet2
--region=us-central1 --project=$onprem_proj_id -q

gcloud compute networks delete cloud-vpc --project $cloud_proj_id
gcloud compute networks delete onprem-vpc --project $onprem_proj_id
```