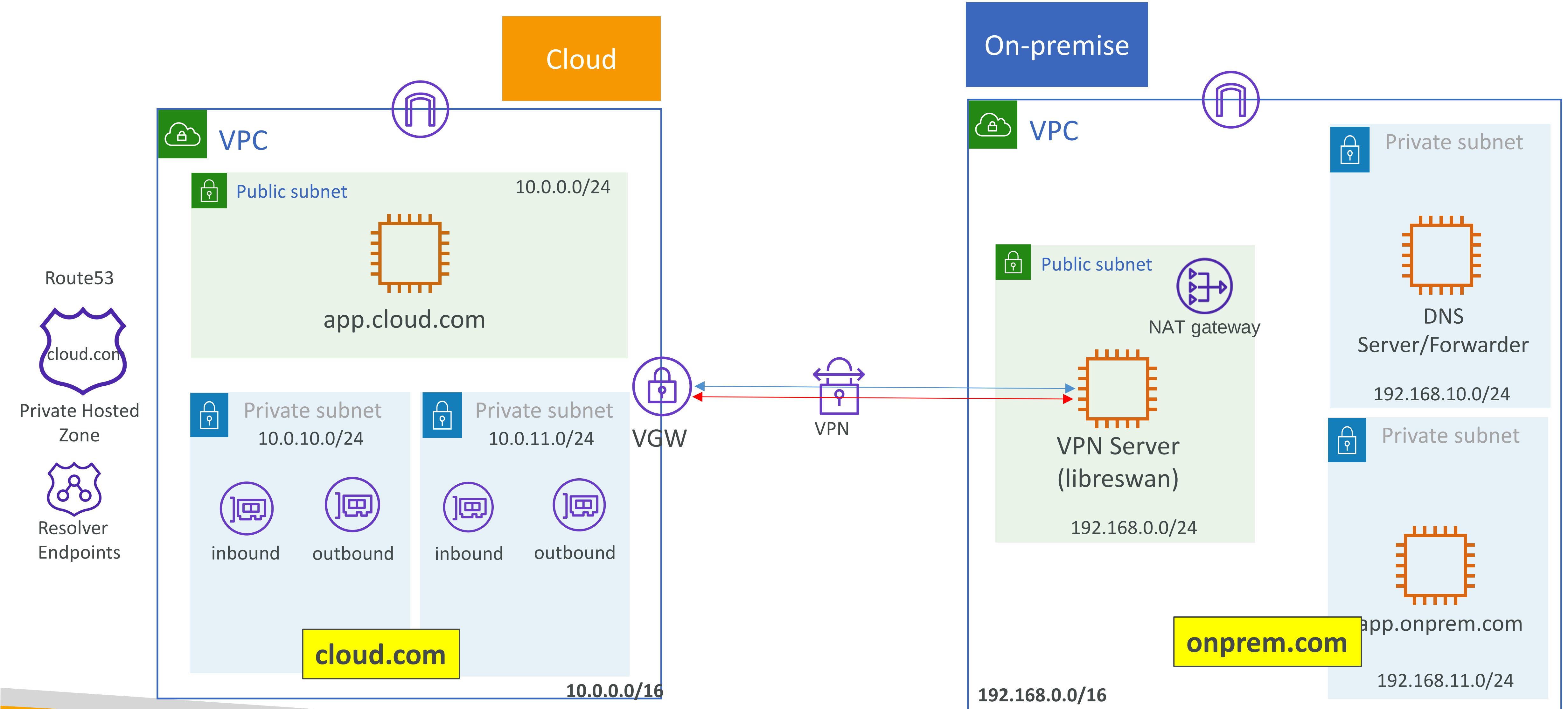
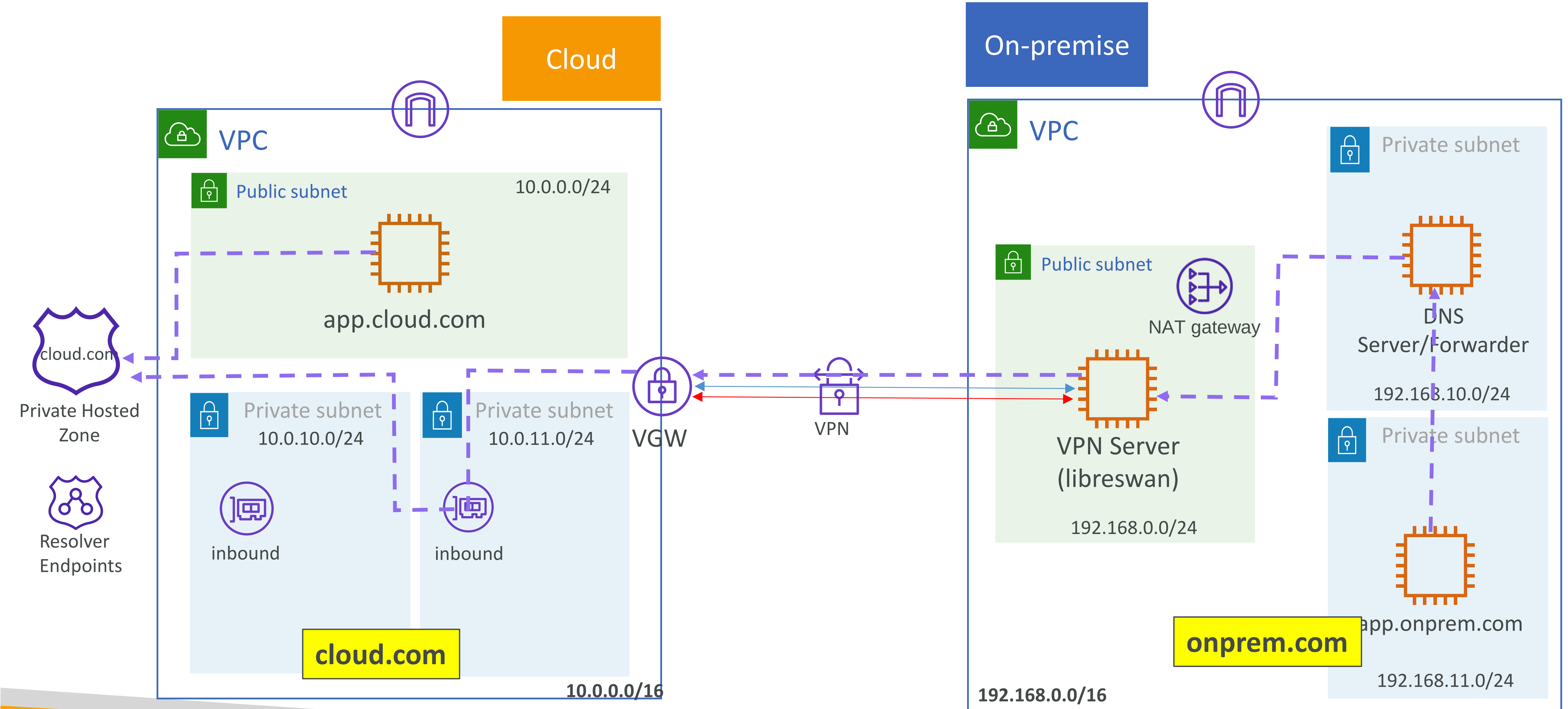
Hands On: Route53 Hybrid DNS

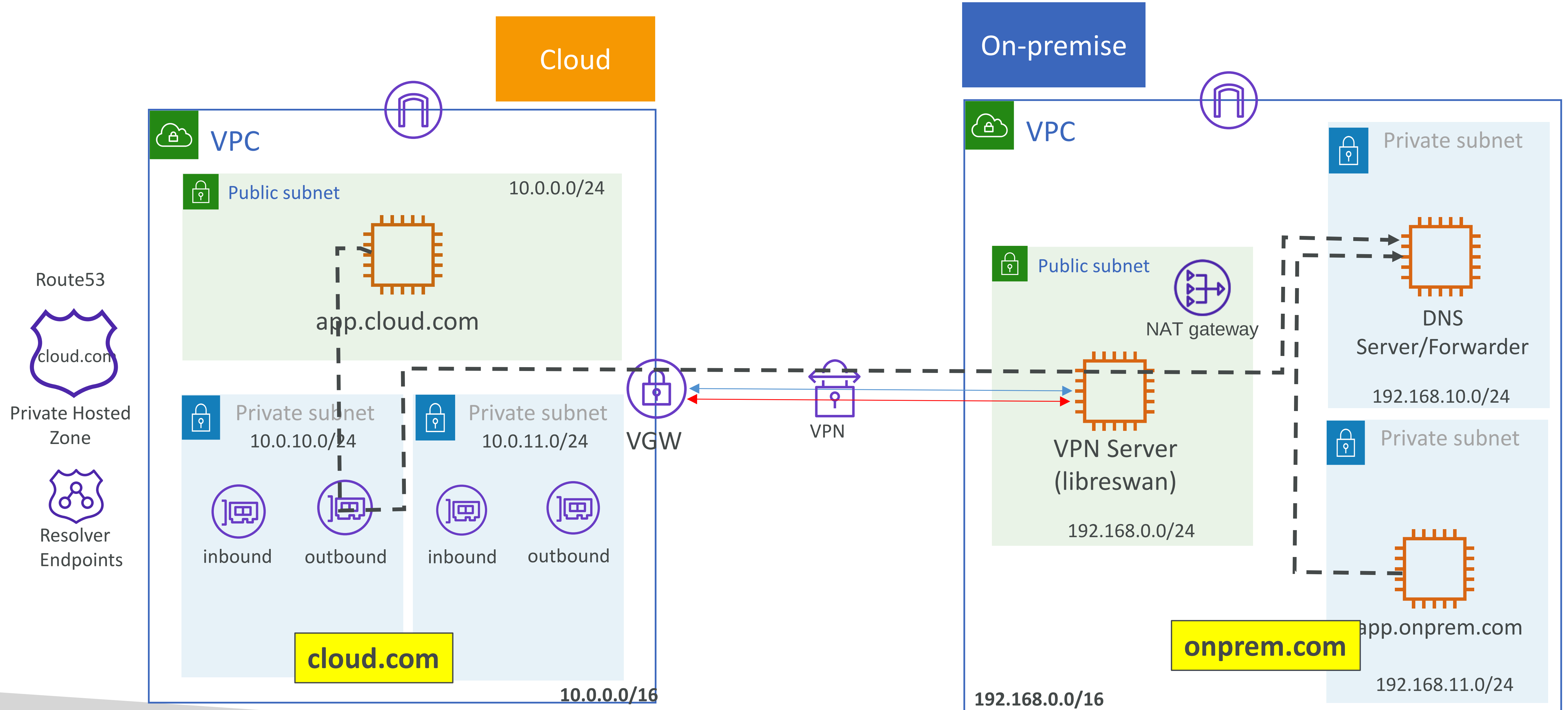
Lab architecture



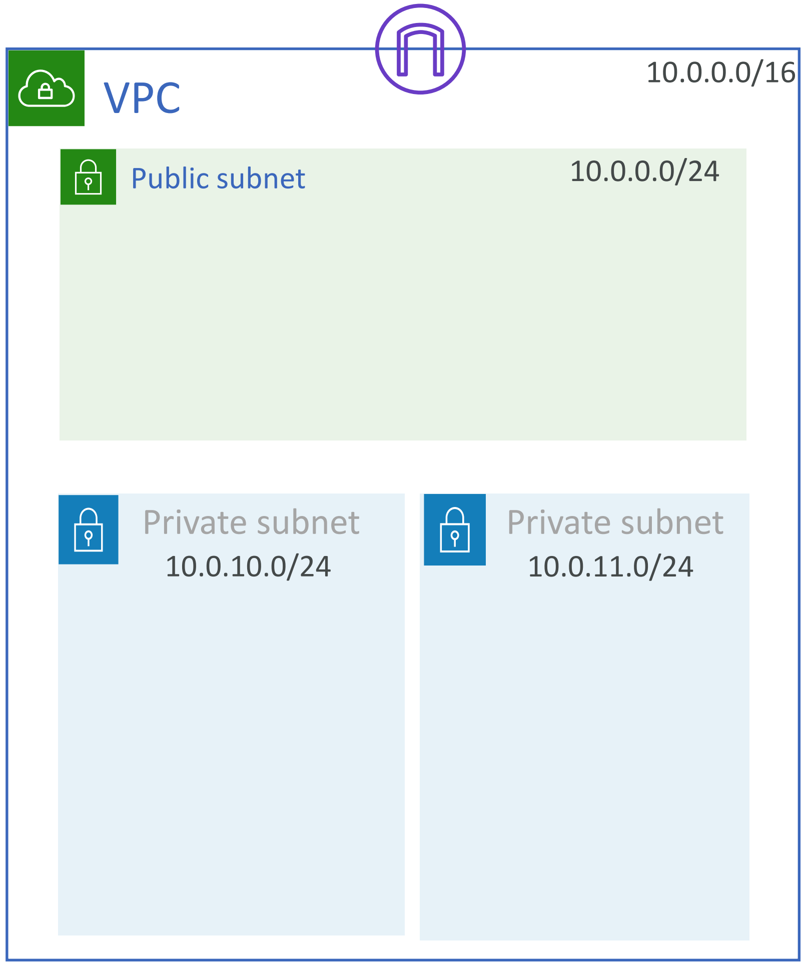
Lab Architecture



Lab Architecture



Setup basic cloud VPC network



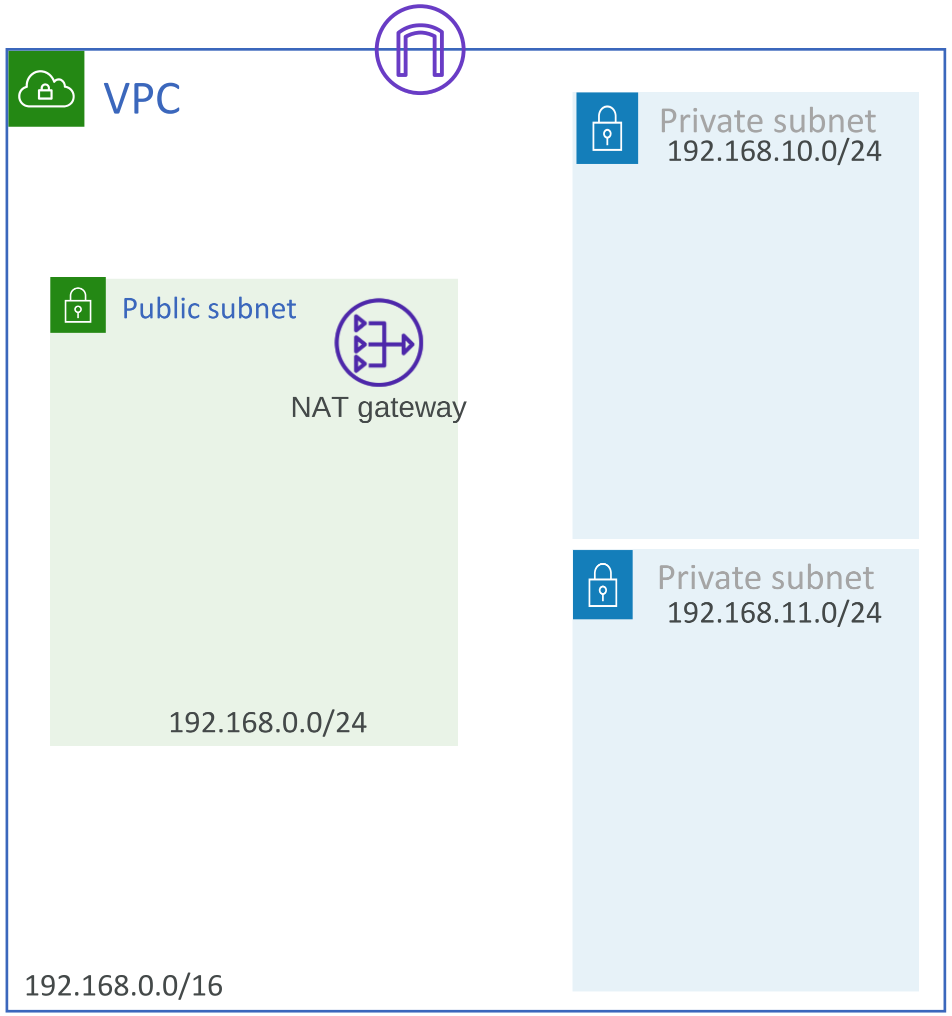
1. Create VPC (10.0.0.0/16), Internet Gateway, 1 Public subnet and 2 private subnets as shown
2. Create route tables for Public and Private subnets.

Pre-created to save some time

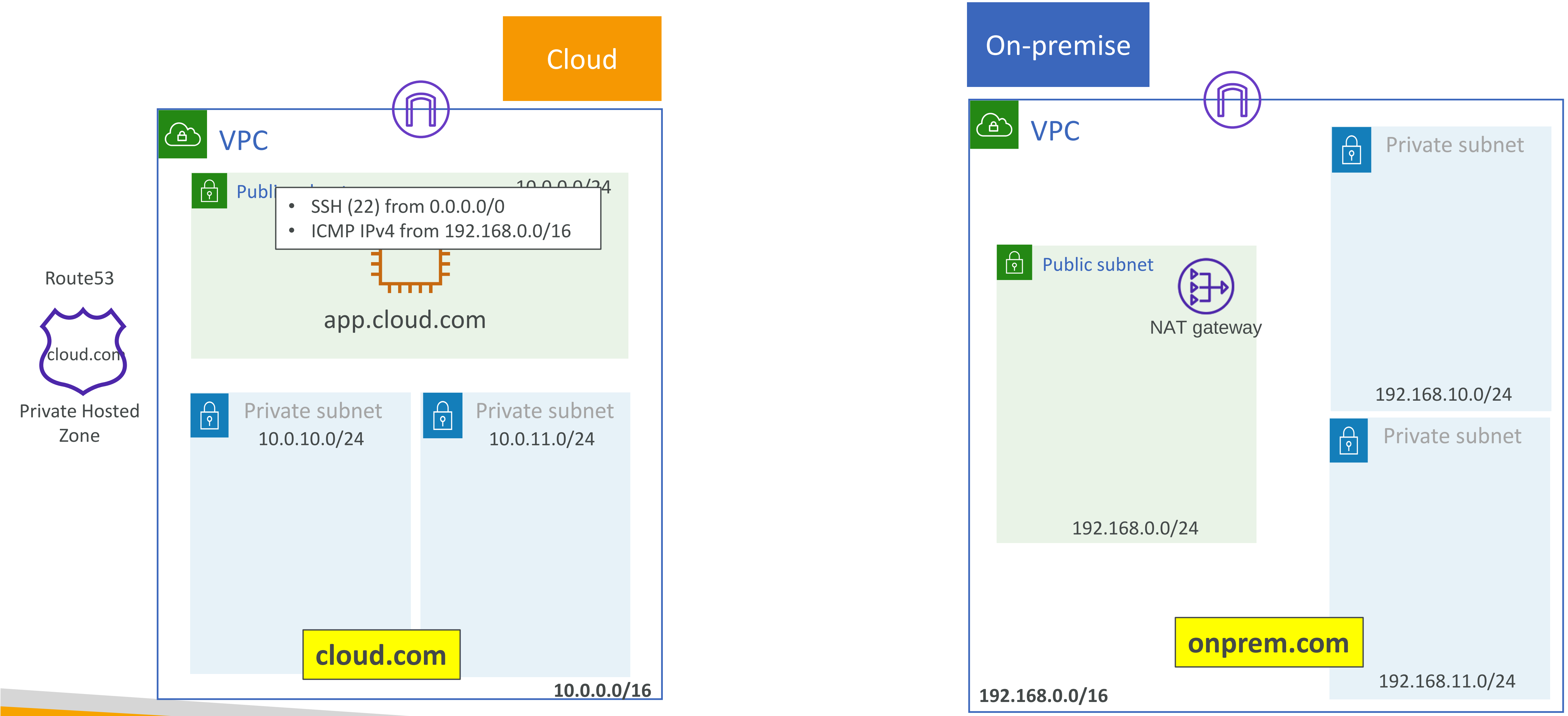
Setup basic on-premises network

1. Create VPC (192.168.0.0/16), Internet Gateway, 1 Public subnet and 2 private subnets as shown
2. Create NAT gateway in Public subnet
3. Create route tables for Public and Private subnets.
4. Update private subnet Route table to route traffic for 0.0.0.0/0 through the NAT gateway.

Pre-created to save some time

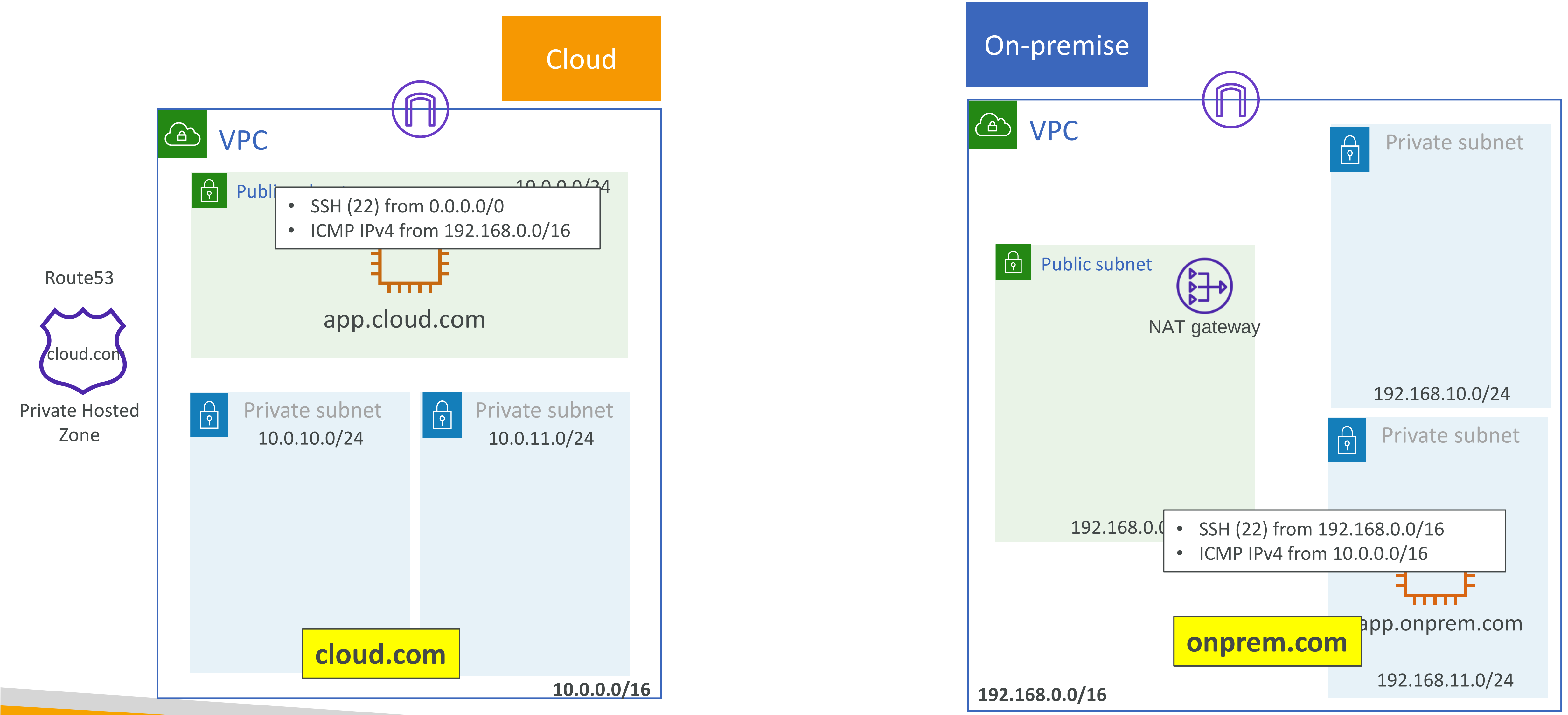


Step 1.1 - Launch app server on Cloud side



- SSH (22) from 0.0.0.0/0
- ICMP IPv4 from 192.168.0.0/16

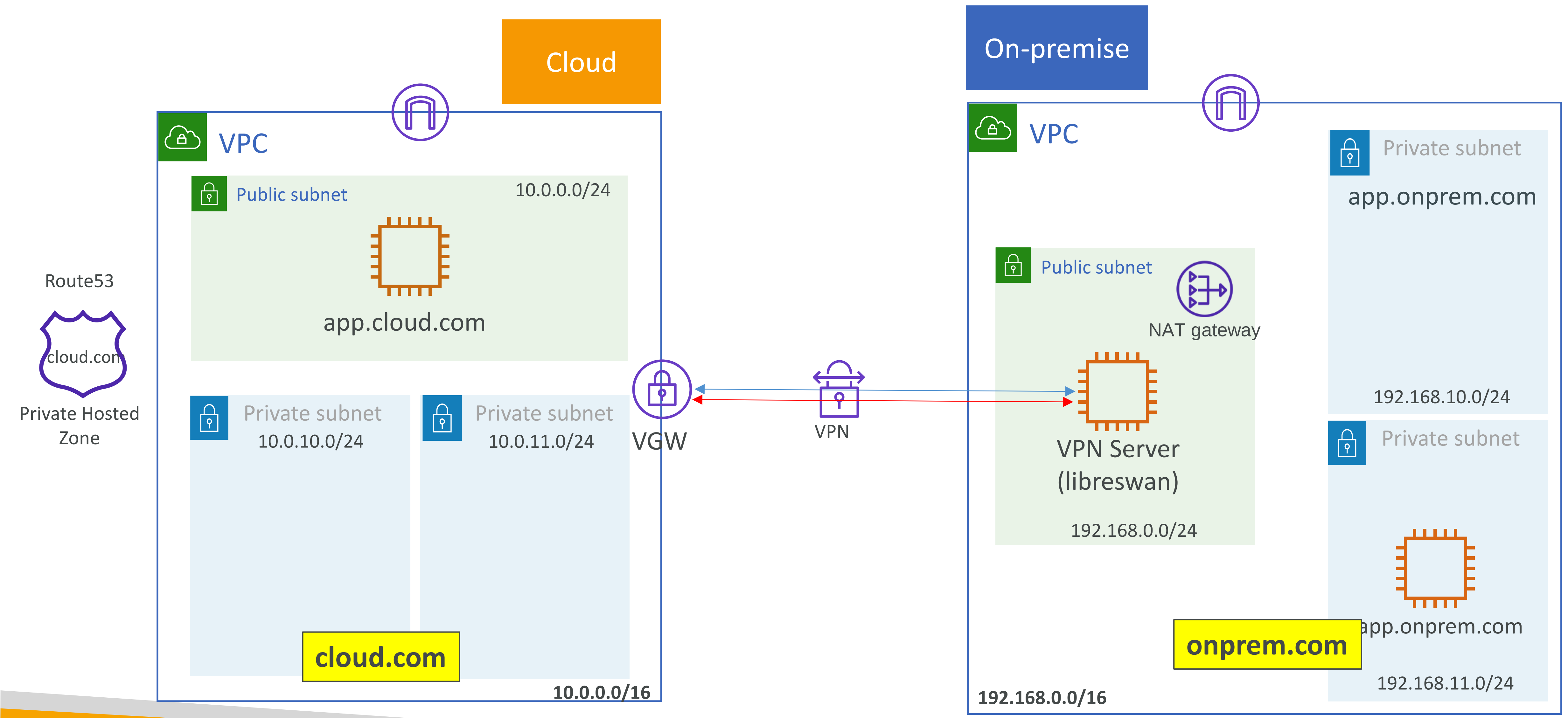
Step 1.2 - Launch app server on on-premise side



- SSH (22) from 0.0.0.0/0
- ICMP IPv4 from 192.168.0.0/16

- SSH (22) from 192.168.0.0/16
- ICMP IPv4 from 10.0.0.0/16

Setup Site-to-Site VPN



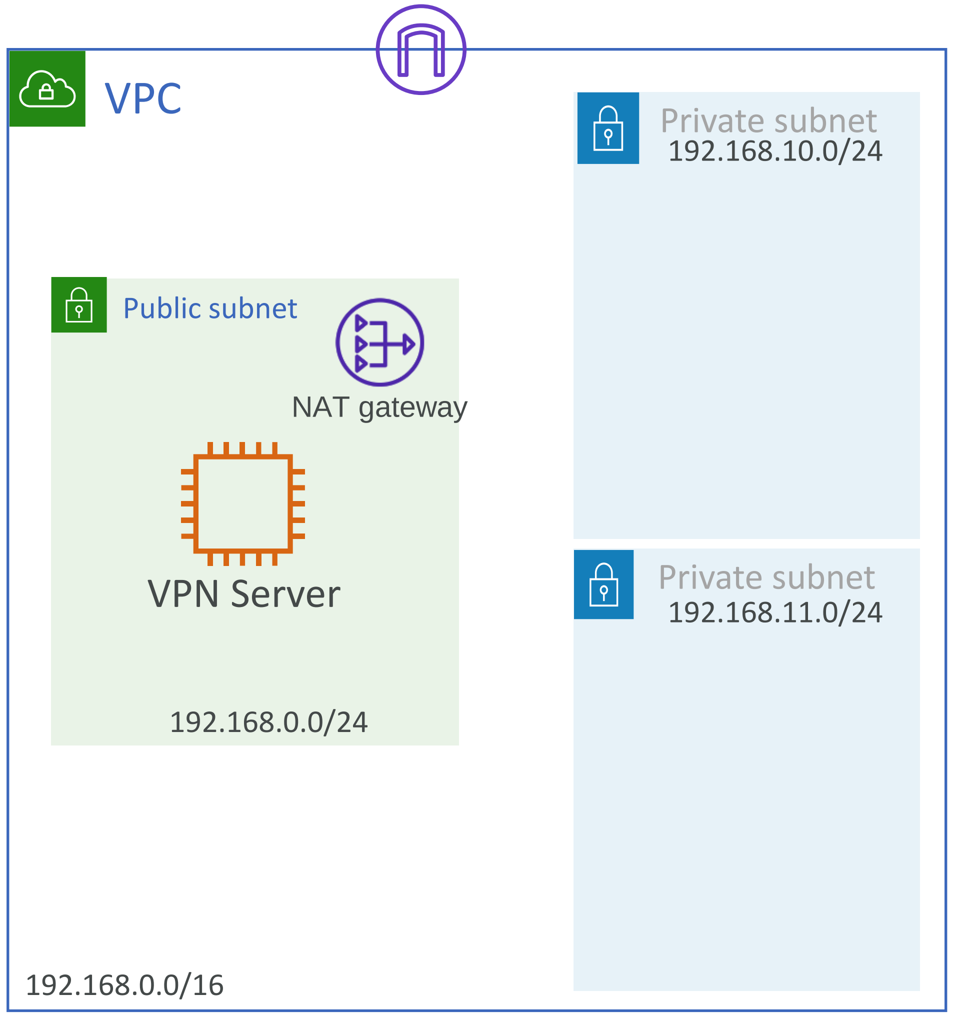
Step 1.3 – Create on-premise VPN server

On-premise

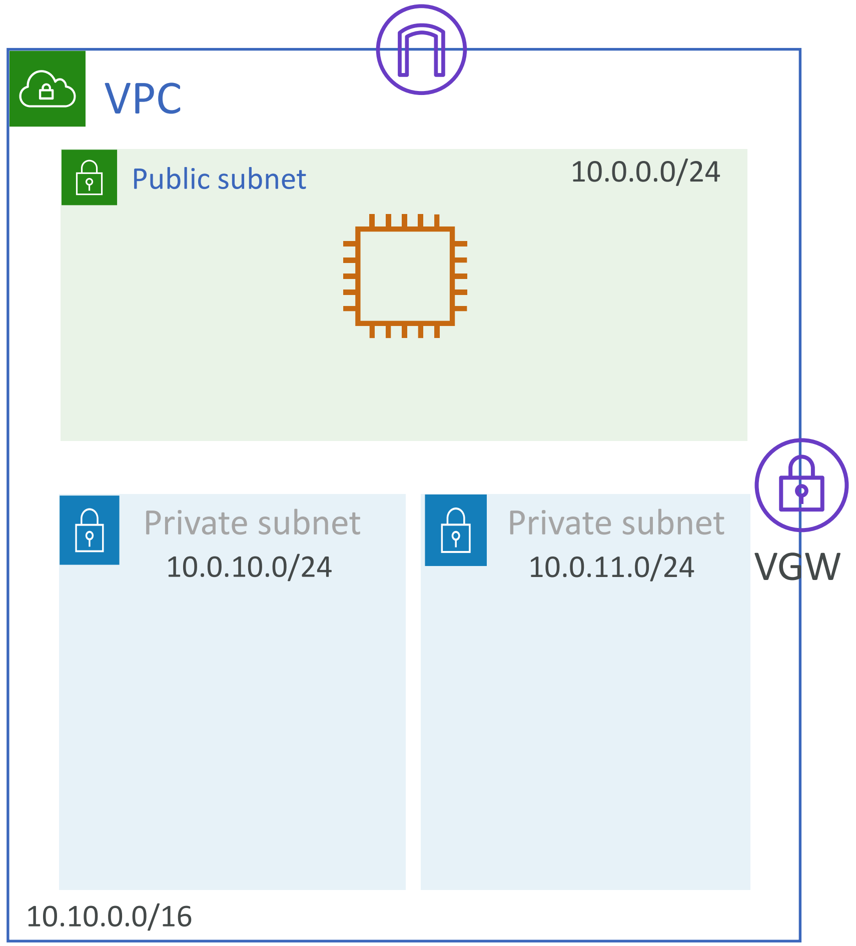
1. Launch EC2 instance in a public subnet (VPN server)
2. Security group of VPN server to allow
   i. SSH from 0.0.0.0/0 (to connect and configure)
   ii. To send the ICMP and DNS traffic from on-prem network to cloud network, open additional ports as below:
      - ICMP IPv4 from 192.168.0.0/16
      - DNS UDP 53 from 192.168.0.0/16

***Note**: If VPN handshake is initiated from the other end, then you should have following inbound rules in VPN server security group. In our case, it's initiated by Libreswan server and SG by default has outbound traffic allowed. Due to SG's statefulness, we don't need to open these ports for inbound traffic. Return traffic is allowed.*

1. *Custom Protocol 50 (ESP) from VGW Public IPs*
2. *UDP 500 from VGW Public IPs*
3. *UDP 4500 from VGW Public Ips if VPN server is behind NAT*



Step 1.4 – Create VPN Connection



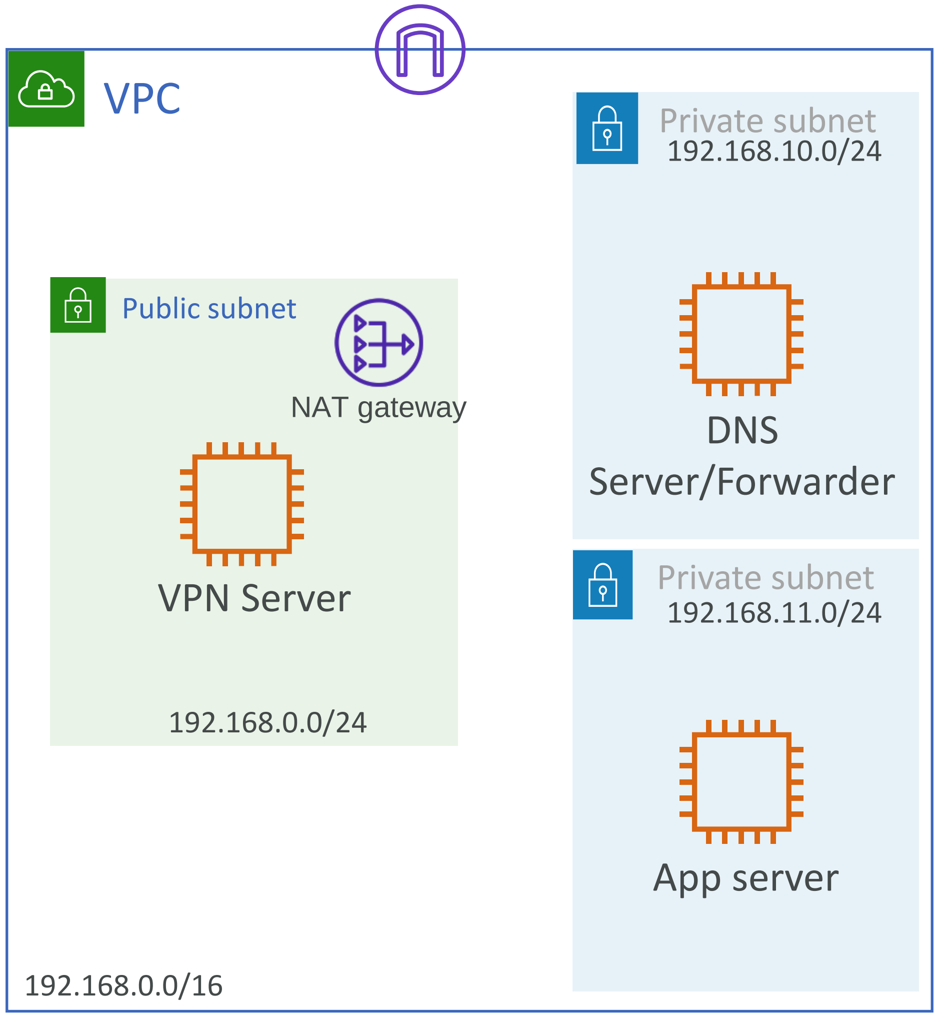
1. Create Virtual Private Gateway and attach to Cloud VPC
2. Create Customer gateway
   1. Use on-premise VPN server EIP
3. Create Site-to-Site VPN connection
   1. Use VGW and CGW created above
   2. Use static routing with IP prefixes – 192.168.0.0/16
   3. Use local IPv4 CIDR as 192.168.0.0/16 (on-prem side) and Remote IPv4 CIDR as 10.0.0.0/16 (Cloud side)
4. Create VPN connection
5. Wait for few minutes and download VPN configuration file for Openswan vendor.

On-premise

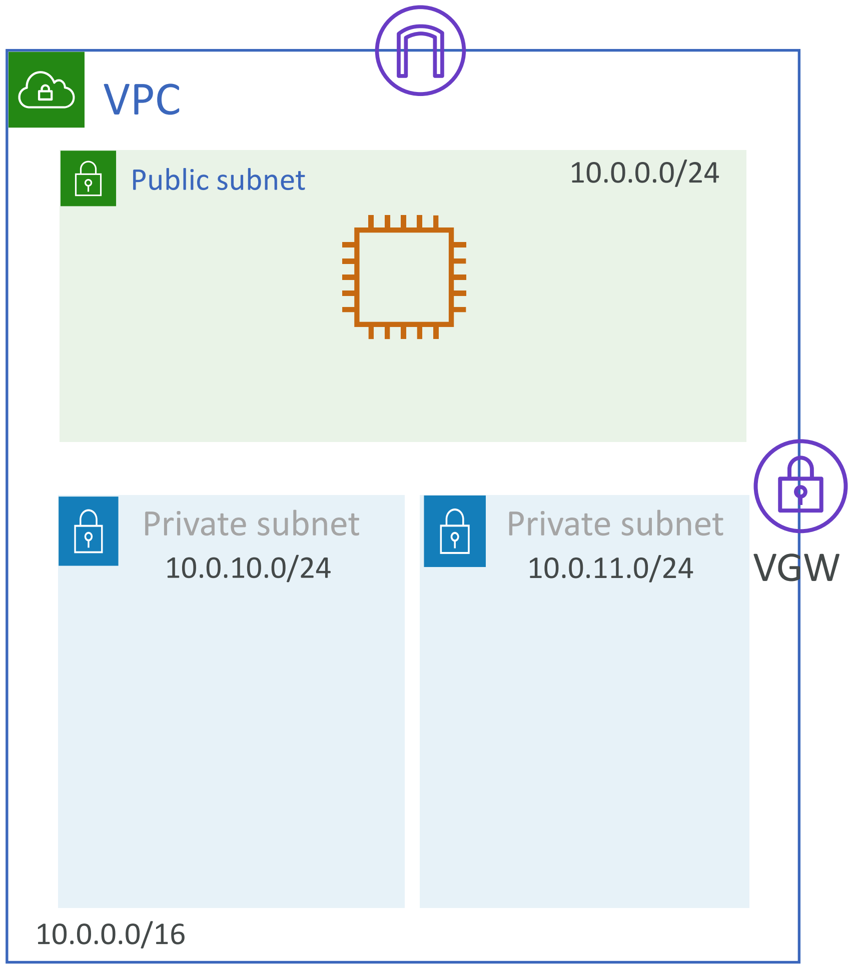
Step 1.5 – Configure on-premise VPN server

1. SSH into the VPN server
2. Install libreswan: *sudo yum install libreswan -y*
3. Open the VPN configuration file you downloaded and follow the instructions in the file to setup **Tunnel 1**
   - ✓ Make sure you remove auth=esp from the configuration
   - ✓ Make sure you change:
     - phase2alg= aes256-sha1;modp2048
     - ike=aes256-sha1;modp2048
4. Start the IPSec service: *sudo systemctl start ipsec.service*
5. Check status of IPSec service: *sudo systemctl status ipsec.service*
6. Go back to AWS console and check the VPN tunnel status –Tunnel 1 should be UP (and Tunnel 2 should be down)

Refer Site-to-Site VPN Section

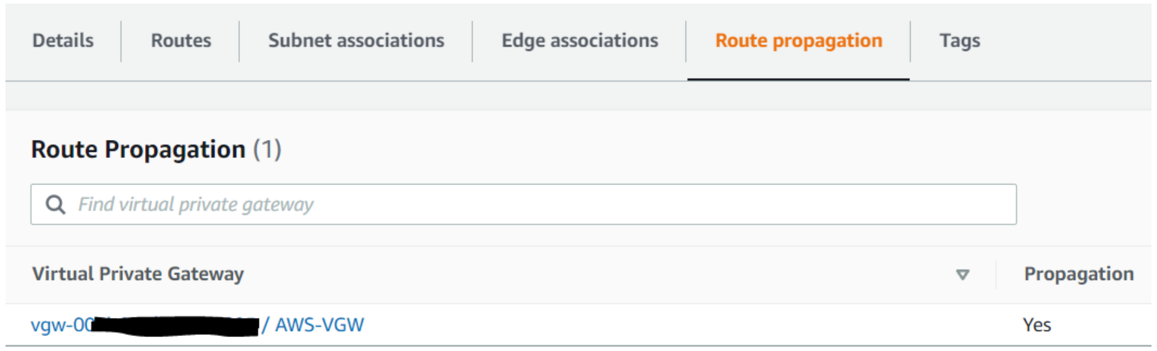


Step 1.6 – Configure routes for Cloud VPC to route VPN traffic through VPN connection



1. Update the Public and Private Subnets route table to propagate routes from the VGW

(This should automatically add a route to the route table for destination 192.168.0.0/16)

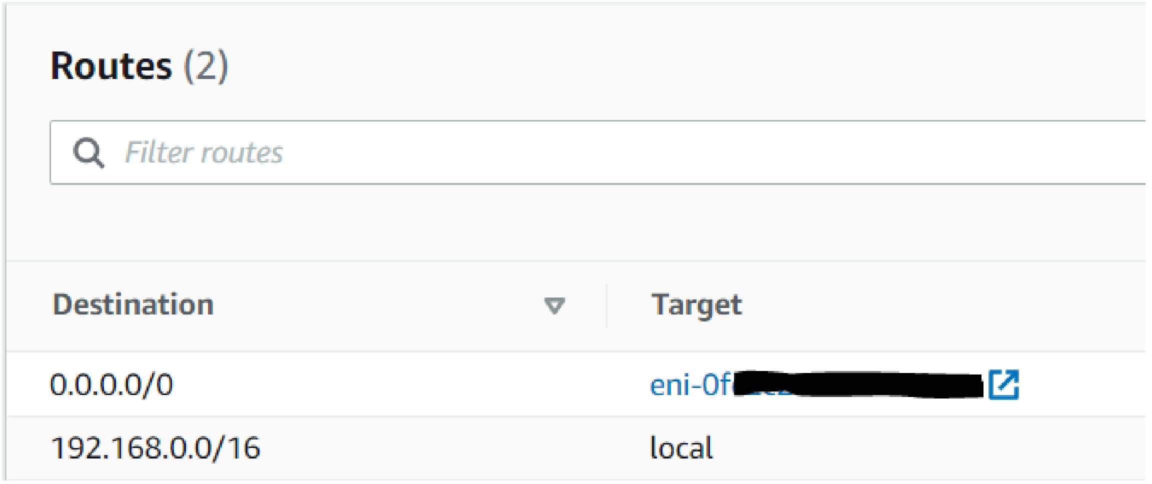


Route Propagation (1)

Virtual Private Gateway	Propagation
vgw-00[redacted]/ AWS-VGW	Yes

Step 1.7 – Configure routes for on-premise to route VPN traffic through VPN server

1. Disable Source/Destination check for VPN server
2. Update the private subnet route table to route the traffic to 10.0.0.0/16 through the VPN server eni.

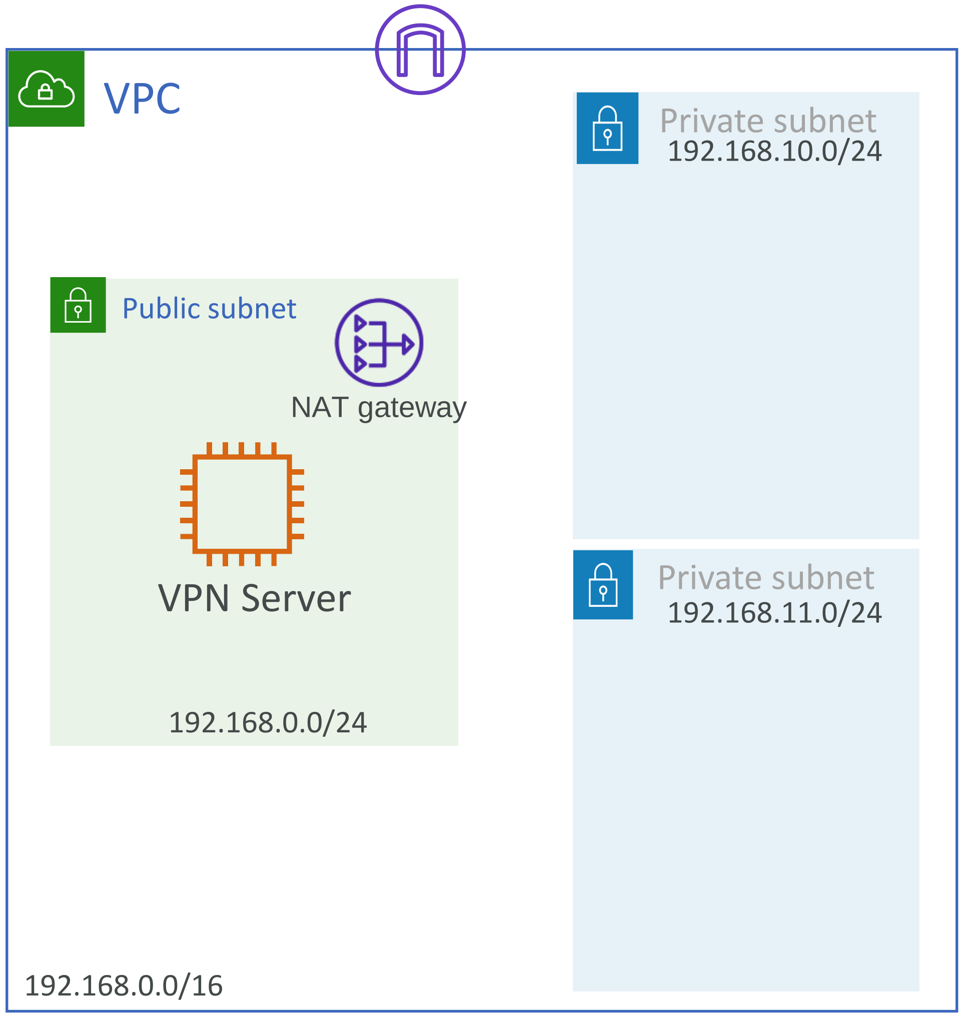


Routes (2)

Filter routes

Destination	Target
0.0.0.0/0	eni-0f[redacted]
192.168.0.0/16	local

Note: Both private subnets are using the same route table.



Step 1.8 – Verify that VPN is working

- From Cloud EC2 instance – Ping to on-premises App server private IP

Cloud EC2 -> VGW -> VPN Tunnel 1 -> VPN server -> App server

- Login to on-premises App server (via SSH into VPN server first and then SSH to app server) – Ping to Cloud EC2 instance private IP

On-prem App server -> On-prem VPN router -> VPN tunnel 1 -> VGW -> Cloud EC2

- From on-premises App server ping public website e.g. google.com. Should be able to reach. This traffic goes out through the NAT gateway.

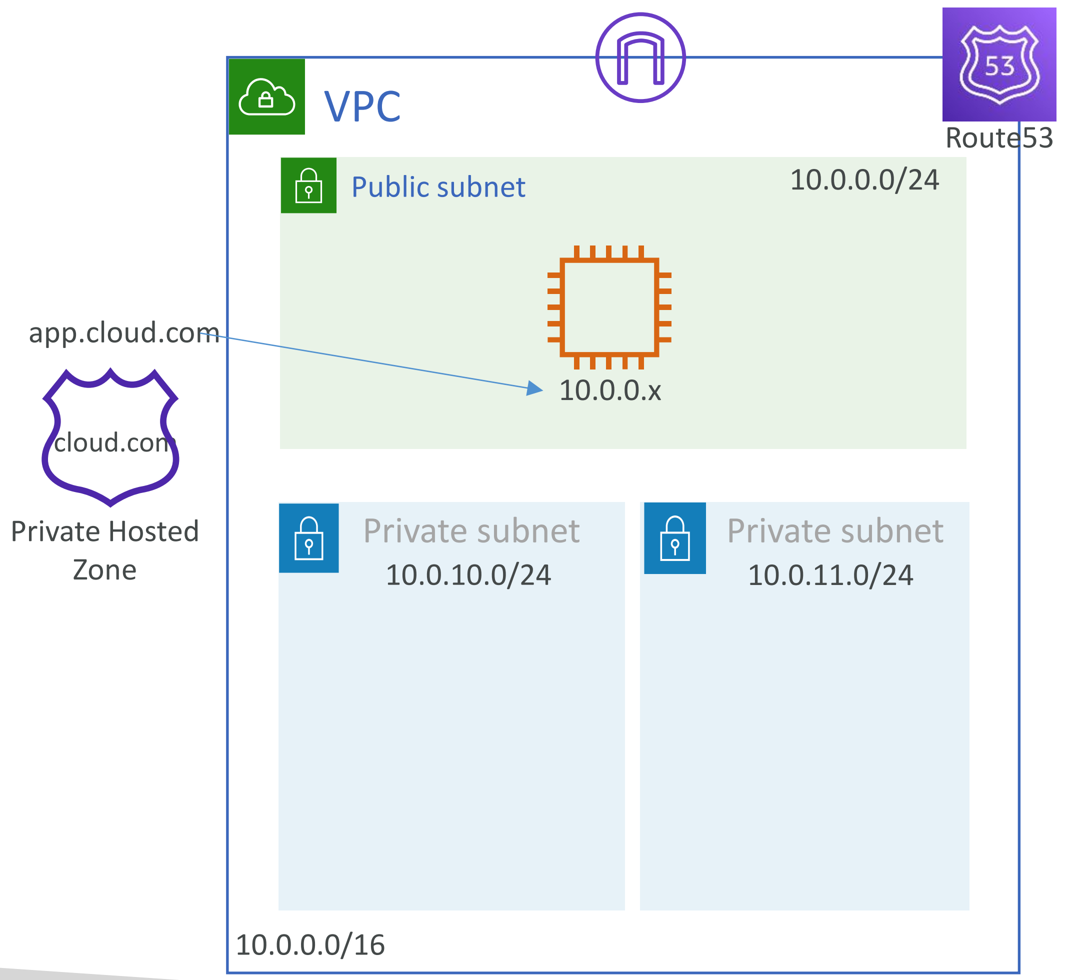
On-prem App server -> NAT gateway -> public website

Well done. VPN working fine.

Step 2 – Configure Cloud and on-premise DNS servers

We will now configure the DNS Server for both cloud and on-premises network

Step 2.1 – Setup cloud DNS using Route53 Private hosted zone



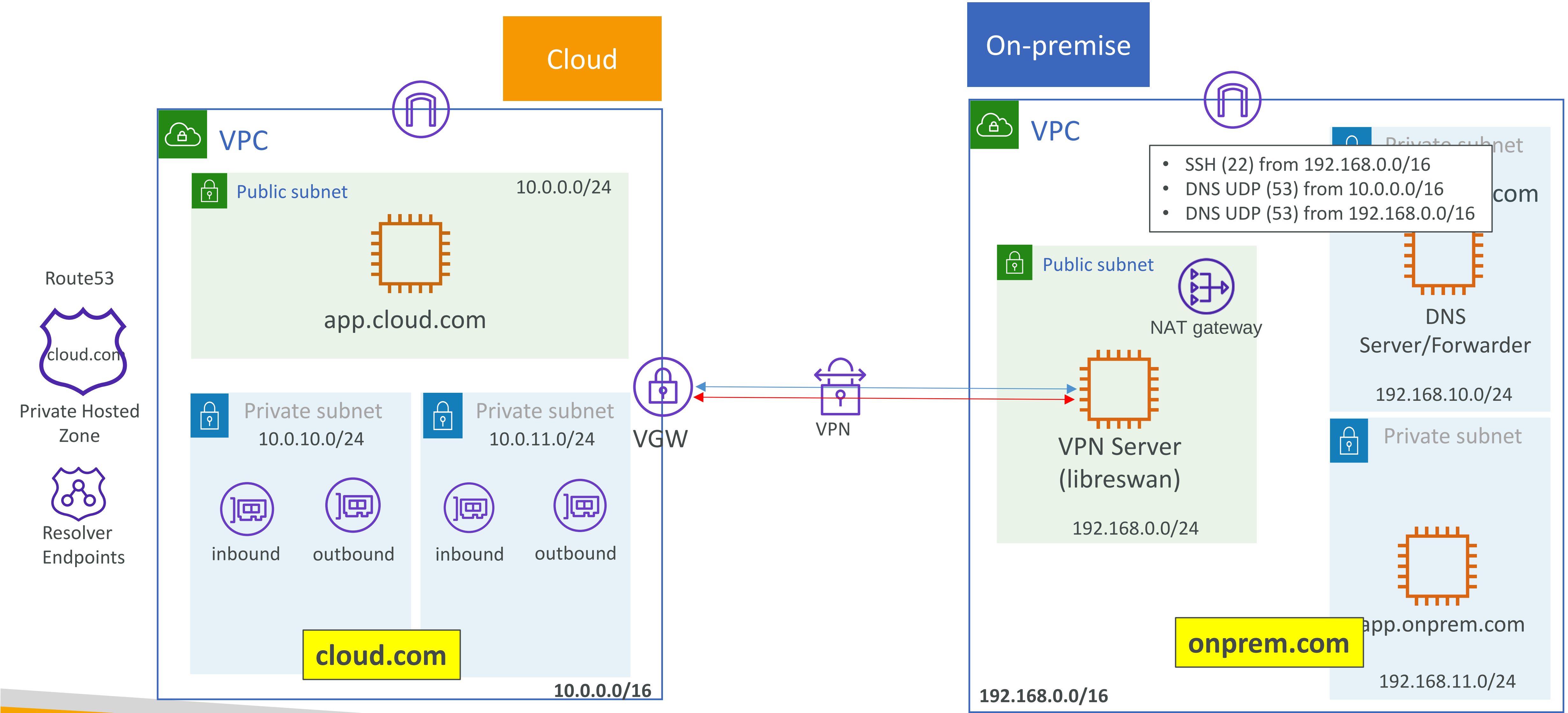
1. For Cloud VPC – **Enable DNS resolution and Enable DNS hostname** (required to use Private hosted zone)
2. Create a Private Hosted Zone (cloud.com) and attach to the Cloud VPC
3. Create A record with name app.cloud.com with Cloud EC2 instance private IP address
4. Login to Cloud EC2 instance and verify if the domain name app.cloud.com is resolving to private IP of itself

```
$nslookup app.cloud.com
```

(may have to wait for couple of minutes)

Step 2.2 - Launch & configure on-premise DNS server

On-premise



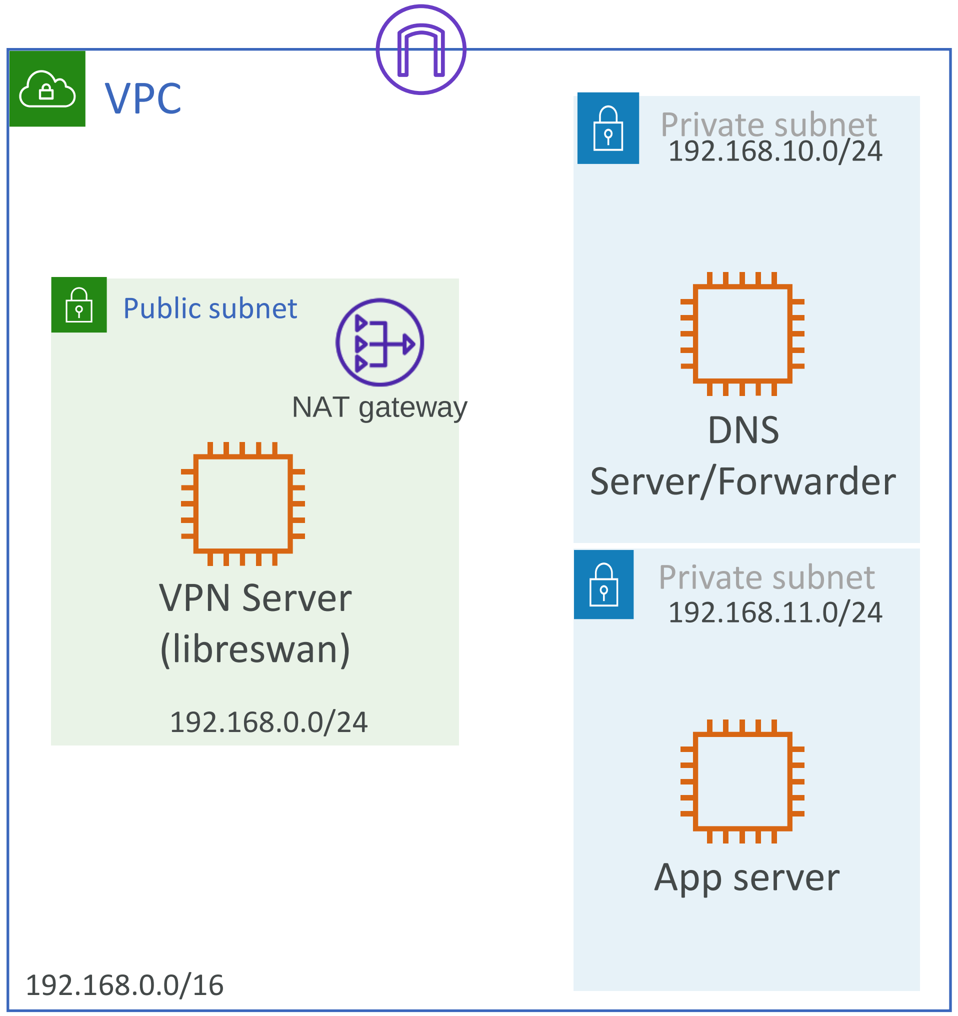
- SSH (22) from 192.168.0.0/16
- DNS UDP (53) from 10.0.0.0/16
- DNS UDP (53) from 192.168.0.0/16

On-premise

Step 2.3 – Setup on-premise DNS server

1. Login to on-premise DNS server (via SSH into VPN server first)
2. Install DNS server packages

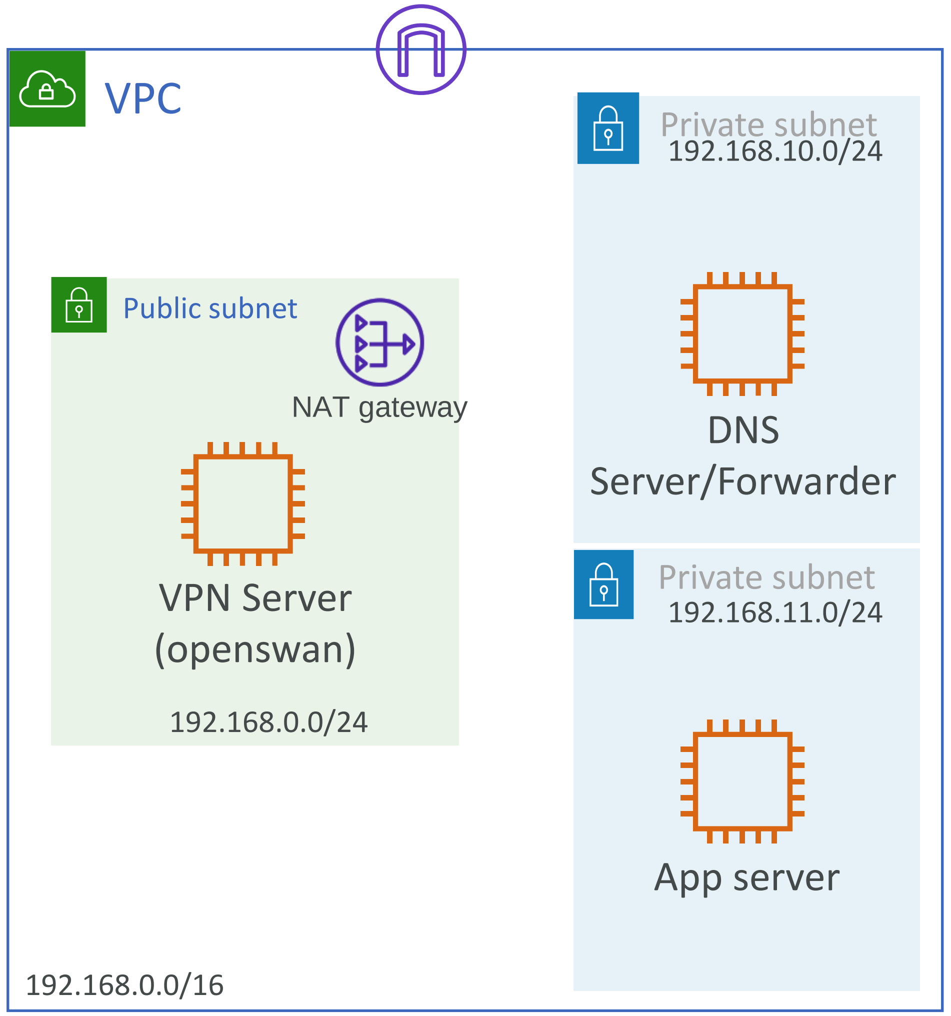
```
$sudo su
$yum update -y
$yum install bind bind-utils -y
```



Step 2.4 – Configure on-premise DNS server

3. Create file /var/named/onprem.com.zone [Replace X.X with your App server IP]

```
$TTL 86400
@   IN  SOA    ns1.onprem.com. root.onprem.com. (
                 2013042201  ;Serial
                 3600        ;Refresh
                 1800        ;Retry
                 604800      ;Expire
                 86400       ;Minimum TTL
)
; Specify our two nameservers
  IN  NS  dnsA.onprem.com.
  IN  NS  dnsB.onprem.com.
; Resolve nameserver hostnames to IP, replace with your two droplet IP addresses.
dnsA   IN  A   1.1.1.1
dnsB   IN  A   8.8.8.8
; Define hostname -> IP pairs which you wish to resolve
@     IN  A   192.168.X.X
App  IN   A  192.168.X.X
```

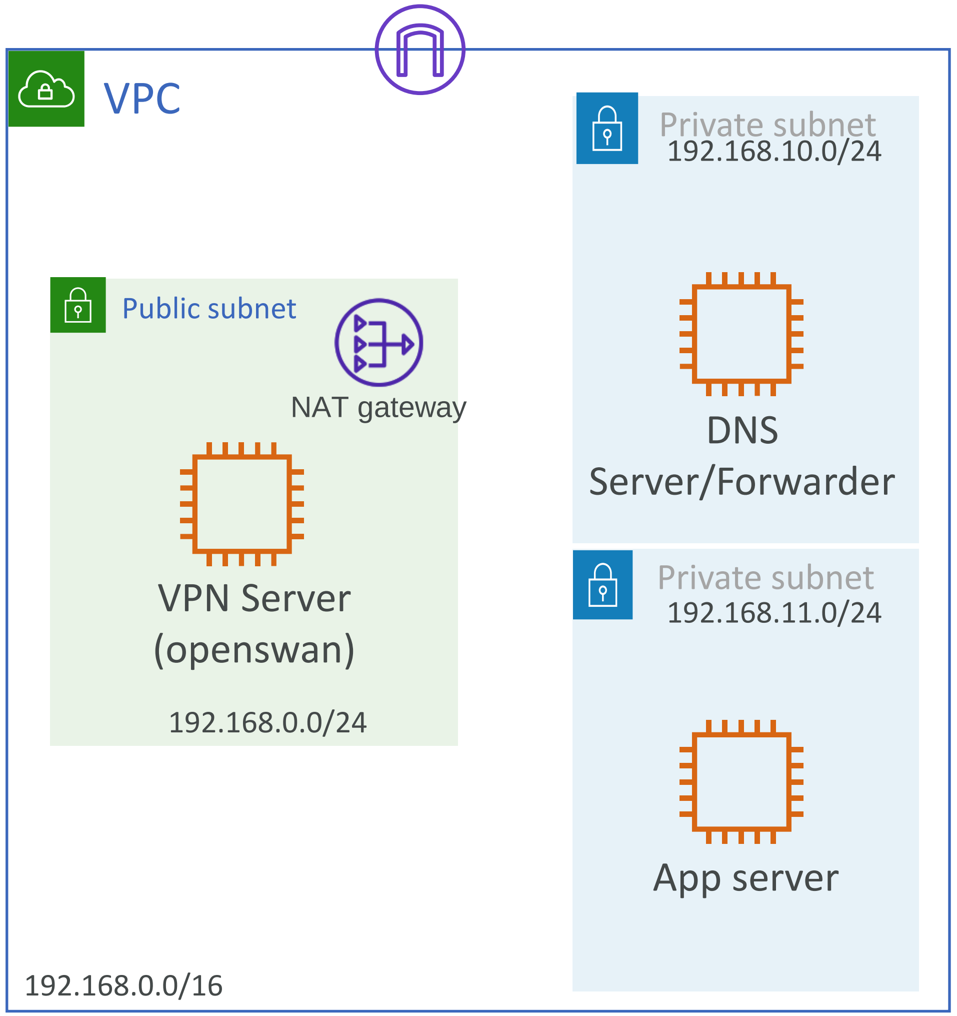


On-premise

Step 2.4 – Configure on-premise DNS server

4. Create file /etc/named.conf [Replace X.X with your DNS server IP]

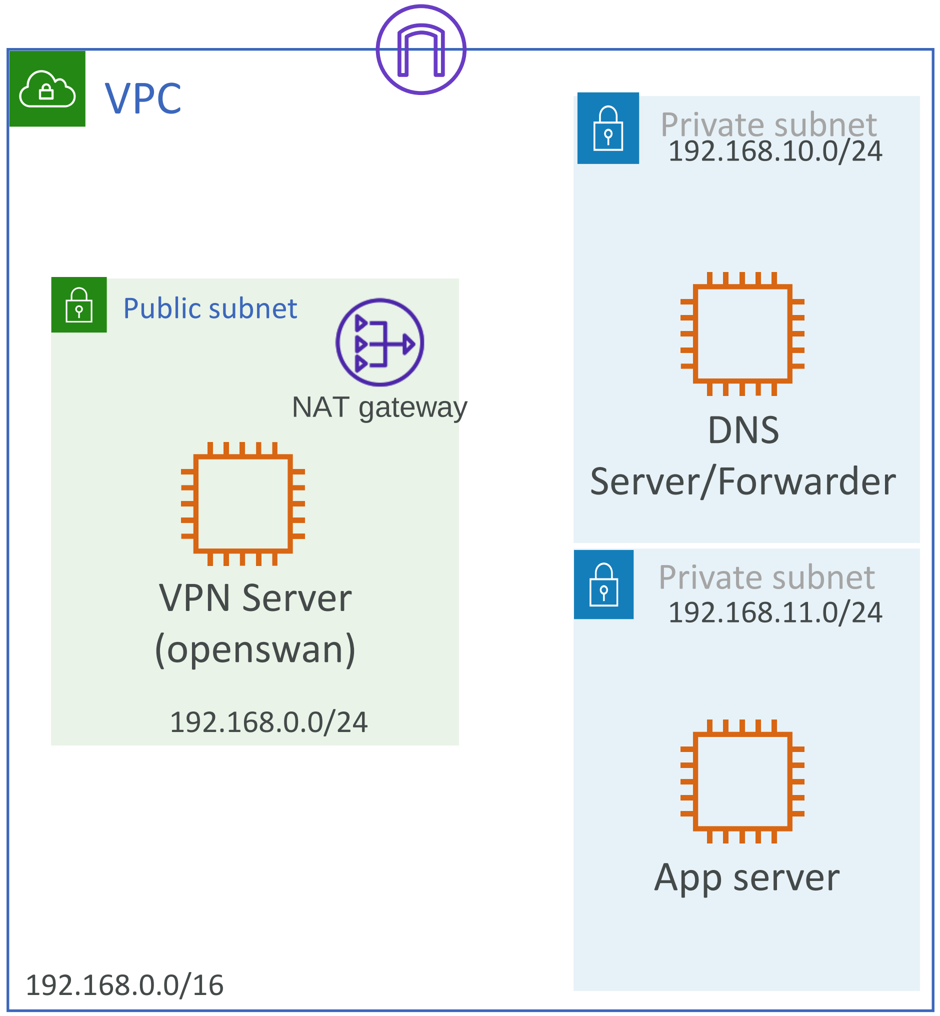
```
options {
  directory "/var/named";
  dump-file "/var/named/data/cache_dump.db";
  statistics-file "/var/named/data/named_stats.txt";
  memstatistics-file "/var/named/data/named_mem_stats.txt";
  allow-query { any; };
  allow-transfer     { localhost; 192.168.X.X; };
  recursion yes;
  forward first;
  forwarders {
    192.168.0.2;
  };
dnssec-validation yes;
/* Path to ISC DLV key */
  bindkeys-file "/etc/named.iscdlv.key";
  managed-keys-directory "/var/named/dynamic";
};
zone "onprem.com" IN {
    type master;
    file "onprem.com.zone";
    allow-update { none; };
};
```



Step 2.4 – Configure on-premise DNS server

5. Restart **named** service

```
$systemctl restart named.service
$chkconfig named on
```



Step 2.5 – Configure App server to send all DNS requests to DNS server

You need to configure the App server to send the DNS queries to you DNS server

1. Add the DNS server details to resolv.conf

nameserver <IP address of on-premises VPN server>

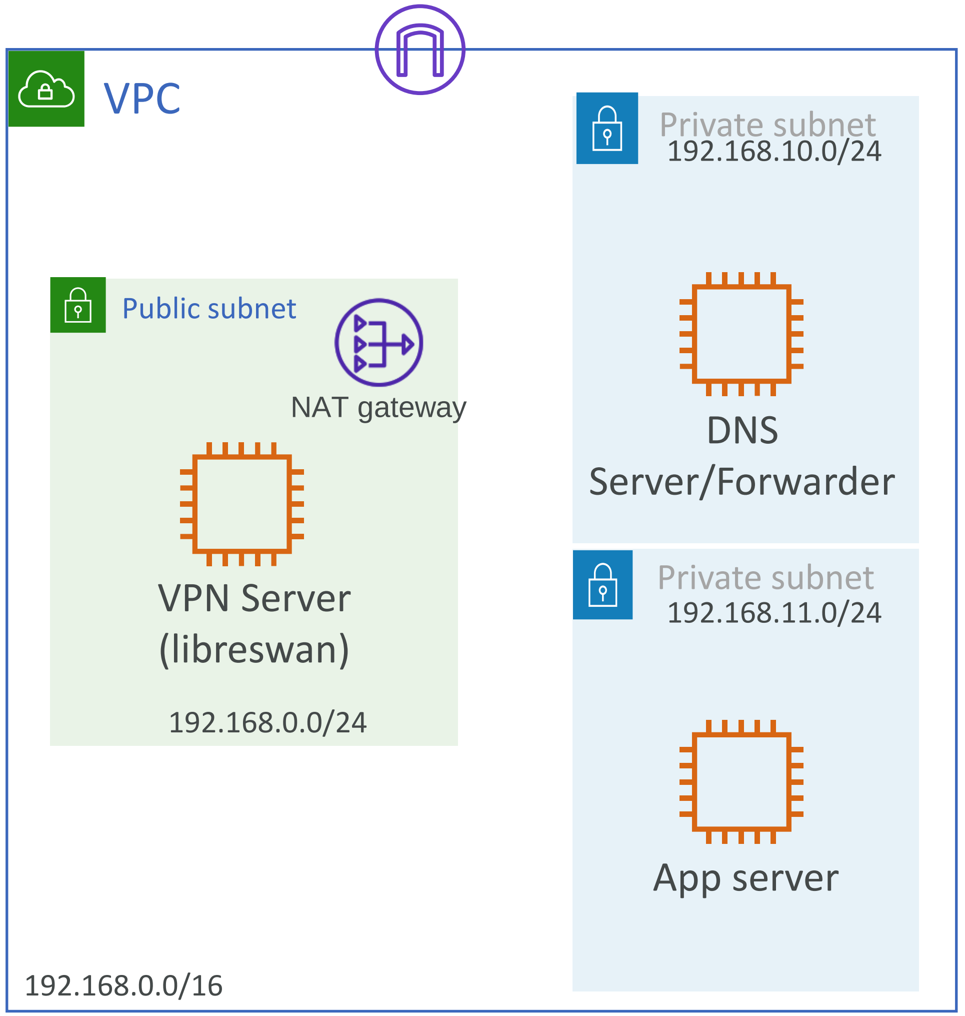
2. Login to App server again and verify that you are able to ping to app.onprem.com

$ping app.onprem.com

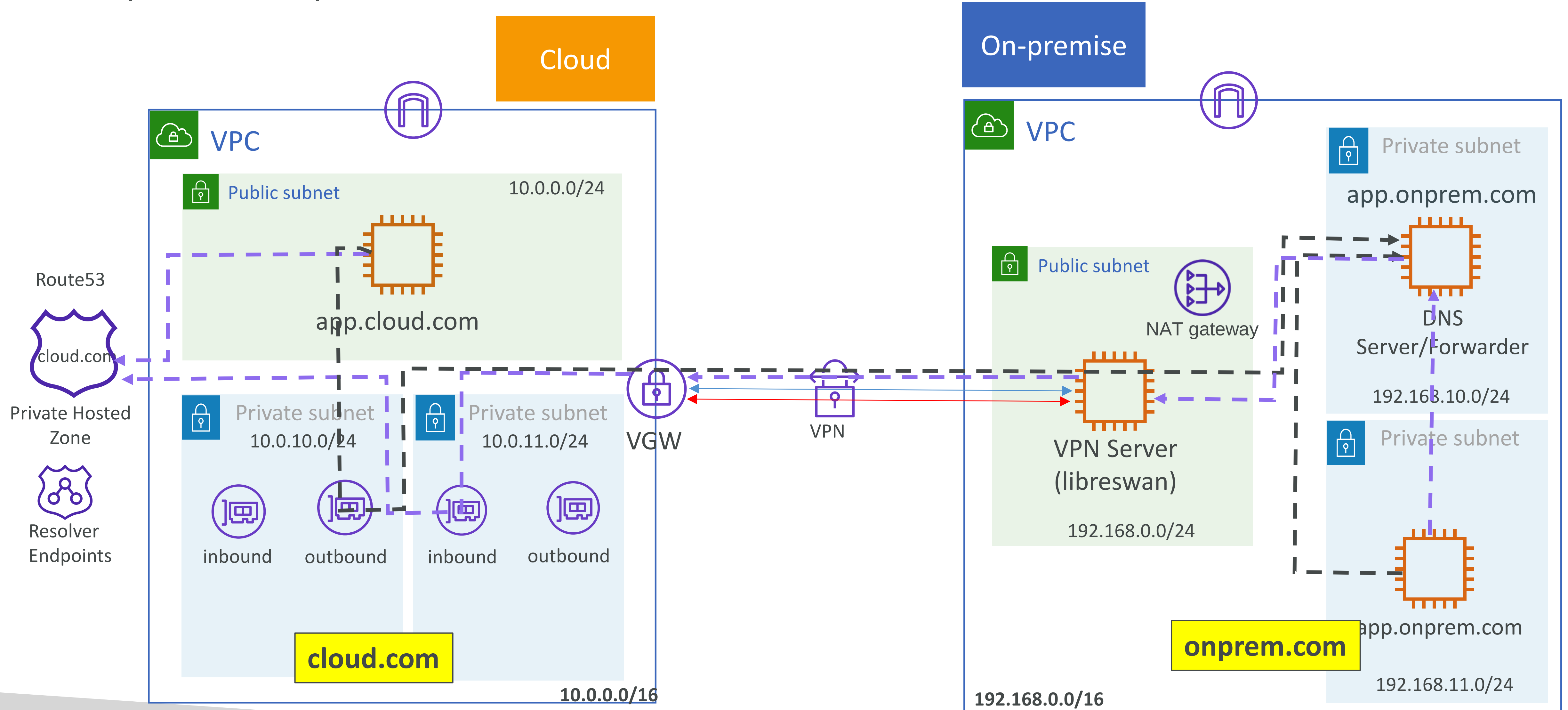
*[At this moment, you **can not** ping to app.cloud.com from on-premise app server and vise-a-versa for obvious reason that we haven't configured hybrid DNS so far]*

Well done. Separate DNS configurations done for Cloud and On-premises environments.

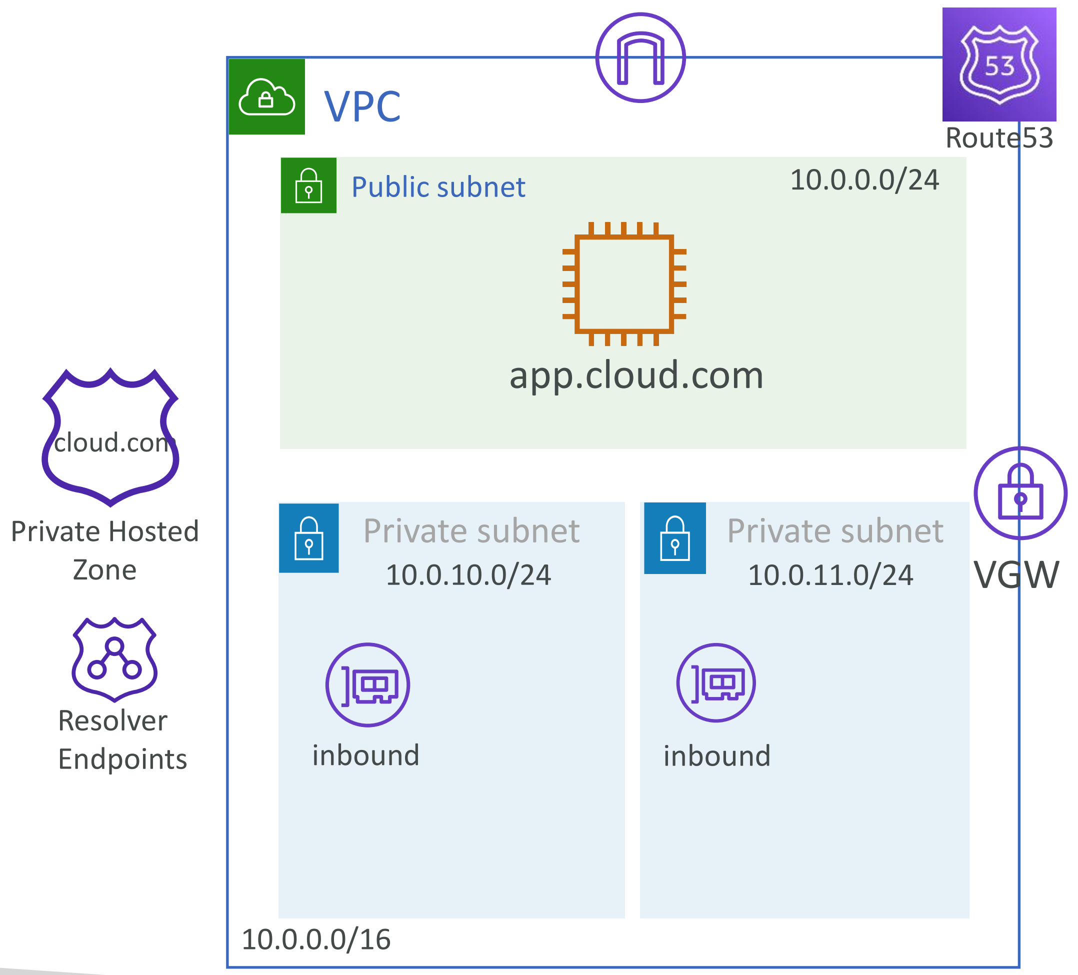
On-premise



Step 3 – Hybrid DNS resolution in both the directions



Step 3.1 Hybrid DNS – From on-premises to AWS



Create Route53 Resolver inbound endpoint

1. Create a Security group for Route53 resolver inbound endpoint
   - Allow DNS (UDP 53) from on-premise network 192.168.0.0/16
2. Create Route53 Resolver inbound endpoint
   1. Make sure you choose same region
   2. Select Cloud VPC
   3. Select 2 private subnets (across 2 AZs) created earlier.

Step 3.2 – Hybrid DNS – From on-premises to AWS

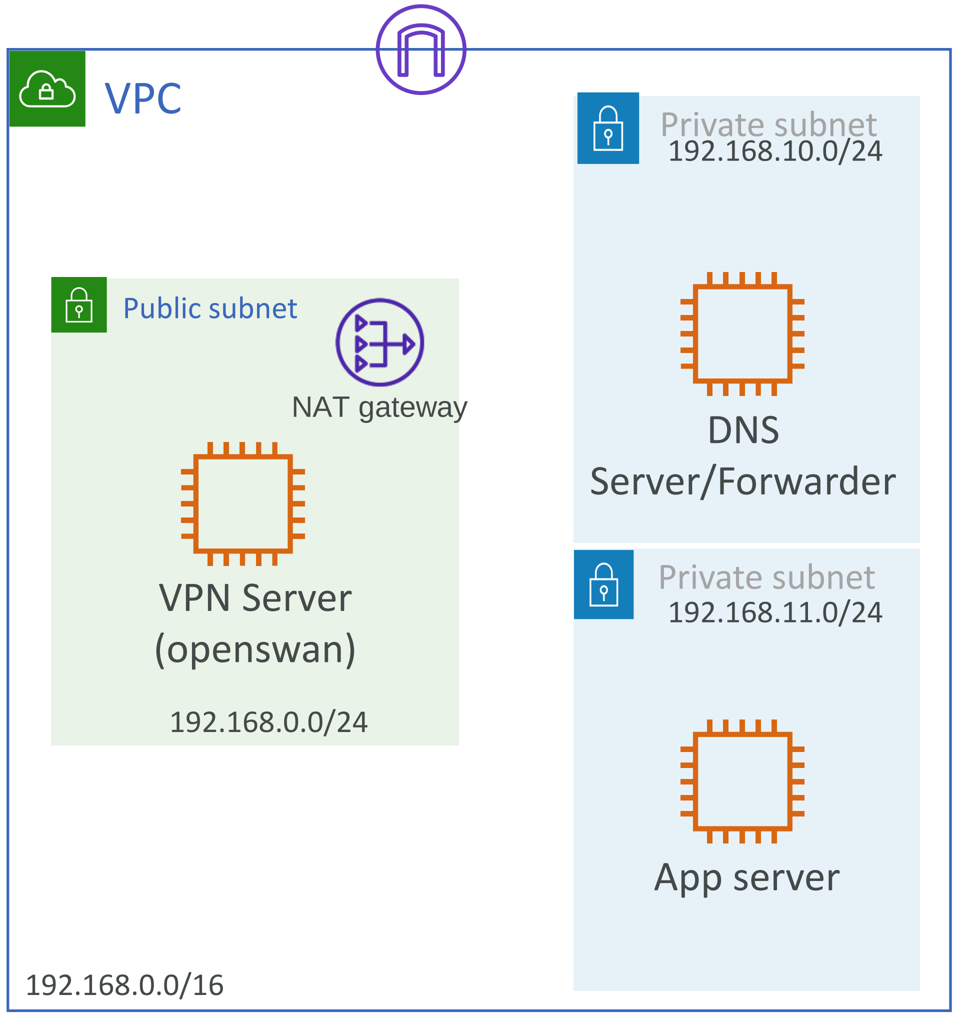
Configure on-premise DNS server to forward DNS queries for cloud.com to Route53 inbound resolver endpoint IPs

1. Add following to /etc/named.conf. Replace ENDPOINT IPs with Route53 inbound resolver IPs.

```
zone "cloud.com" {
  type forward;
  forward only;
  forwarders { INBOUND_ENDPOINT_IP1; INBOUND_ENDPOINT_IP2; };
};
```

2. Restart named service

$systemctl restart named.service



On-premise

Step 3.3 – Verify DNS resolution

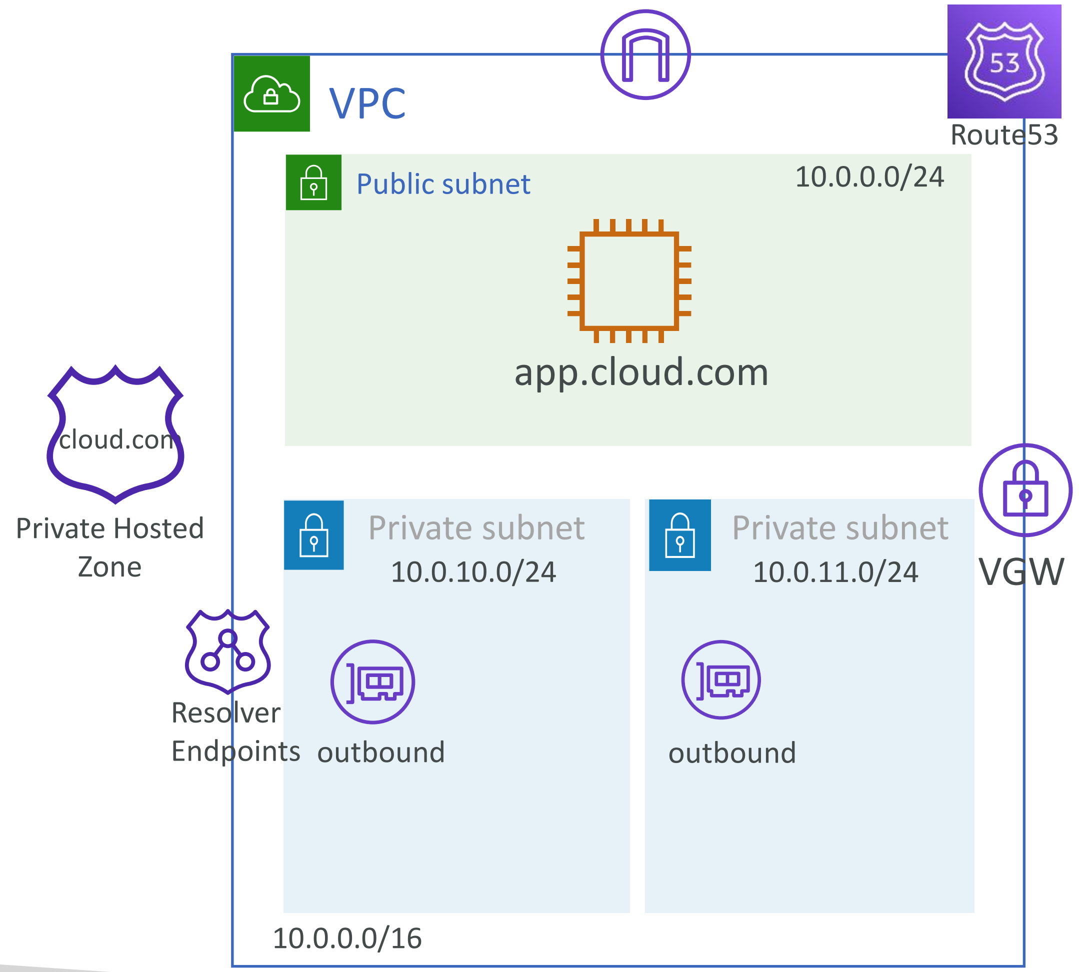
You should be able to resolve app.cloud.com from the on-premise App Server

1. From on-premise App Server

```
$nslookup app.cloud.com
$ping app.cloud.com
```

Well done ! On-premises to AWS DNS resolution working as expected.

Step 3.4 – Hybrid DNS – From cloud to on-premises (outbound)



Create Route53 Resolver outbound endpoint

1. Create a Security group for Route53 resolver outbound endpoint
   - Outbound Rule: Allow DNS (UDP 53) to on-premise network 192.168.0.0/16
2. Create Route53 Resolver outbound endpoint
   1. Make sure you choose same region
   2. Select Cloud VPC
   3. Select 2 private subnets (across 2 AZs) created earlier.
3. Create Outbound Rule to forward the DNS requests for onprem.com to DNS server IP
   1. Domain name as onprem.com
   2. Select Cloud VPC
   3. Target IP address of on-premises DNS server IP (192.168.X.X)

Step 3.5 – Verify DNS resolution

You should be able to resolve app.onprem.com from the Cloud EC2 instance

1. From Cloud EC2 instance

$nslookup app.onprem.com
$ping app.onprem.com

Amazing ! AWS to on-premises DNS resolution is also working!

Cleanup

- Terminate all EC2 instances created in this lab.
- Delete VPN Connection, VGW and CGW
- Delete NAT Gateway
- Delete Route53 resolver rules followed by resolver endpoints
- Delete Route53 A record followed by a private hosted zone
- Delete both VPCs