Centralized IPv4 and IPv6 Egress using Proxy Instances and NLB

These step by step instructions describe how to setup the Centralized IPv4 and IPv6 Egress using Proxy Instances and NLB solution illustrated in Centralizing outbound Internet traffic for dual stack IPv4 and IPv6. Before proceeding, make sure to complete the steps described in Baseline Architecture. The following diagrams outline the network architecture and the corresponding route tables we're going to setup:

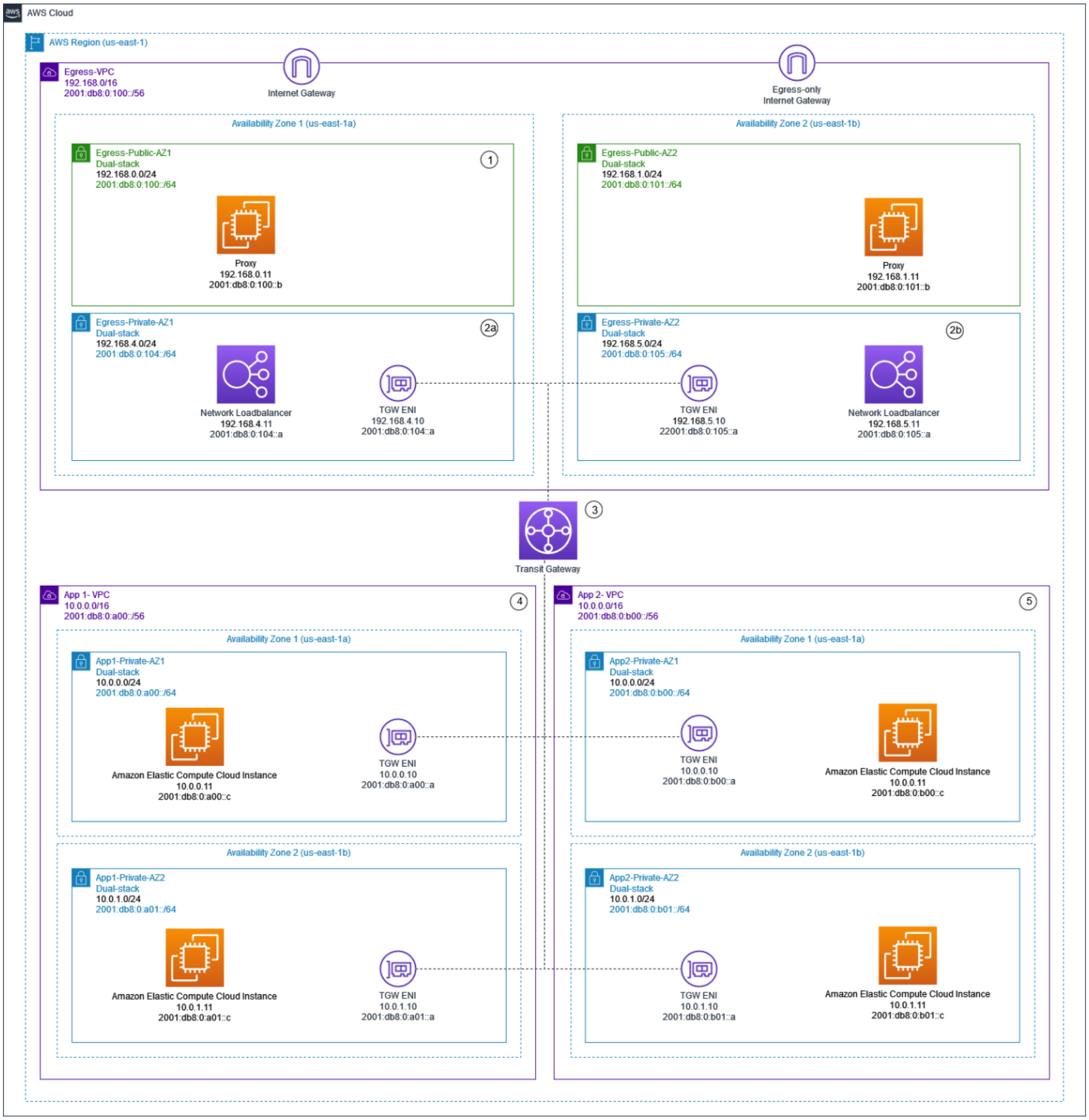
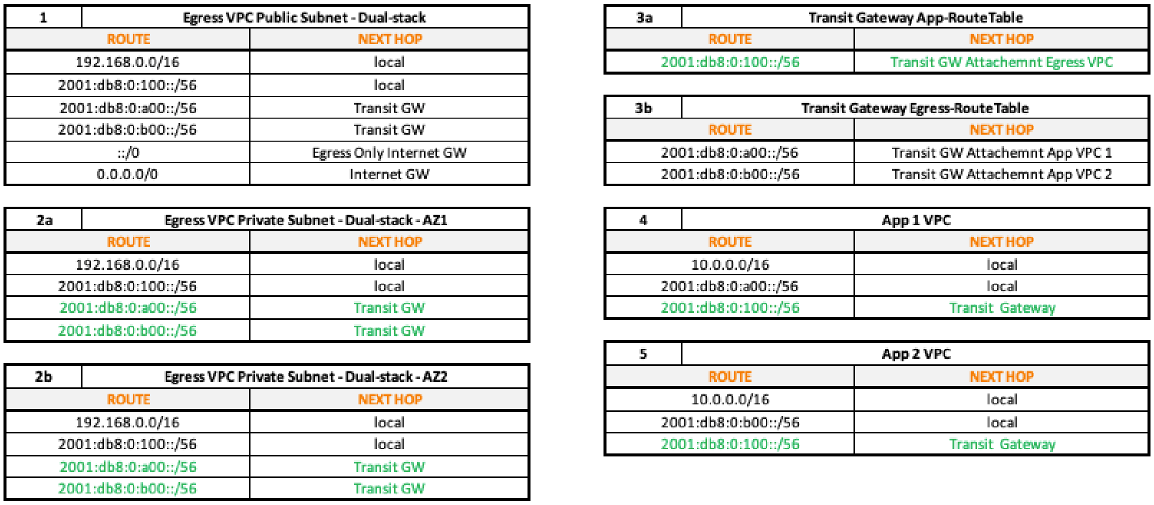


Figure 1: Centralized IPv4 and IPv6 Egress using Proxy Instances and NLB



1	Egress VPC Public Subnet - Dual-stack
ROUTE	NEXT HOP
192.168.0.0/16	local
2001:db8:0:100::/56	local
2001:db8:0:a00::/56	Transit GW
2001:db8:0:b00::/56	Transit GW
::/0	Egress Only Internet GW
0.0.0.0/0	Internet GW

2a	Egress VPC Private Subnet - Dual-stack - AZ1
ROUTE	NEXT HOP
192.168.0.0/16	local
2001:db8:0:100::/56	local
2001:db8:0:a00::/56	Transit GW
2001:db8:0:b00::/56	Transit GW

2b	Egress VPC Private Subnet - Dual-stack - AZ2
ROUTE	NEXT HOP
192.168.0.0/16	local
2001:db8:0:100::/56	local
2001:db8:0:a00::/56	Transit GW
2001:db8:0:b00::/56	Transit GW

3a	Transit Gateway App-RouteTable
ROUTE	NEXT HOP
2001:db8:0:100::/56	Transit GW Attachemnt Egress VPC

3b	Transit Gateway Egress-RouteTable
ROUTE	NEXT HOP
2001:db8:0:a00::/56	Transit GW Attachemnt App VPC 1
2001:db8:0:b00::/56	Transit GW Attachemnt App VPC 2

4	App 1 VPC
ROUTE	NEXT HOP
10.0.0.0/16	local
2001:db8:0:a00::/56	local
2001:db8:0:100::/56	Transit Gateway

5	App 2 VPC
ROUTE	NEXT HOP
10.0.0.0/16	local
2001:db8:0:b00::/56	local
2001:db8:0:100::/56	Transit Gateway

Figure 2: Route Tables configuration for Centralized IPv4 and IPv6 Egress using Proxy Instances and NLB

Egress VPC Setup

1. Add 2 new routes in the route table Egress-Public-RT, one with the destination 2001:db8:0:100::/56 and the other with the destination 2001:db8:0:a00::/56. Associate the routes with the TGW-Internet.

Application VPCs and Transit Gateway Setup

1. Choose AWS Transit Gateway Route tables and select App-RouteTable. Choose Routes, Create route, enter the 2001:db8:0:100::/56 route, and choose the attachment: Egress-Attachment.
2. In the left navigation pane, choose Route Tables and edit the default route tables associated with App1-VPC and App2-VPC, adding a 2001:db8:0:100::/56 route and set TGW-Internet as the target.

Proxy Instances Setup

1. Create a Target group with the following characteristics (where not specified you can leave the default values. For more information, see Create a target group):
    a. Target type: Instances.
    b. Target group name: proxy-tg.
    c. Protocol: TCP.
    d. Port: 3128.
    e. VPC: Egress-VPC.
    f. During Register Targets step you can avoid registering any instance.
2. Create a Network Load Balancer (NLB) with the following characteristics (where not specified you can leave the default values. For more information, see Create a Network Load Balancer):

    a. Load balancer name: proxy-lb.
    b. Scheme: Internal.
    c. IP Address Type: dualstack.
    d. VPC: Egress-VPC.
    e. Mappings: select Egress-Private-AZ1 for AZ1 and Egress-Private-AZ2 for AZ2.
    f. Listener Port: 3128.
    g. Listener Default Action: Forward to proxy-tg.

3. Create a security group with the following characteristics (where not specified you can leave the default values. For more information, see Create a security group):
    a. Security group name: proxy-sg.
    b. Description: Security group for the proxy EC2 instances.
    c. VPC: Egress-VPC.
    d. Inbound rules:
        i. Click on Add rule.
        ii. Specify as Type Custom TCP, Port Range 3128 and Source 192.168.3.0/24 (Egress-Private-AZ1).
        iii. Repeat the process but specify as source 192.168.4.0/24 (Egress-Private-AZ2).
4. Create a Launch configuration with the following characteristics (where not specified you can leave the default values. For more information, see Create your launch template):
    a. Launch template name: proxy-lt.
    b. Auto Scaling guidance: check Provide guidance to help me set up a template that I can use with EC2 Auto Scaling.
    c. Amazon Machine Image (AMI): select Quick Start then Amazon Linux.
    d. Firewall (security groups): select Select existing security group then proxy-sg.
    e. User data (add as many IPv6 ranges as required under the section App VPCs):

```
#!/usr/bin/env bash
#
# Proxy EC2 instance user data script.

# Error messages
trap 'echo "Aborting due to errexit on line $LINENO. Exit code: $?" >&2' ERR

# Strict mode
set -Eeuo pipefail

# Set $IFS to only newline and tab
IFS=$'\n\t'

# Install squid
yum install -y squid

# Configure squid
cat <<EOF > /etc/squid/squid.conf
#
# Recommended minimum configuration:
#
```

```
# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl localnet src fc00::/7       # RFC 4193 local private network range
acl localnet src fe80::/10      # RFC 4291 link-local (directly plugged) machines

acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

# VPCs
acl vpc src 192.168.0.0/16 # Egress VPC

#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost
http_access allow vpc

# And finally deny all other access to this proxy
```

```
http_access deny all

# Squid normally listens to port 3128
http_port 3128

# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256

# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid

#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320
EOF

# Enable squid
systemctl enable squid

# Start squid
systemctl start squid
```

5. Create an Auto Scaling group with the following characteristics (where not specified you can leave the default values. For more information, see Create an Auto Scaling group using a launch template):
   a. Auto Scaling group name: proxy-asg.
   b. Launch template: proxy-lt.
   c. VPC: Egress-VPC.
   d. Availability Zones and subnets: select Egress-Private-AZ1 and Egress-Private-AZ2.
   e. Instance Type requirements: select Manually add instance types, then for Primary instance type select t3.small and insert 1 for weight. Remove all of the automatically added Additional instance types.
   f. Load balancing: Attach to an existing load balancer.
   g. Attach to an existing load balancer: select Choose from your load balancer target groups, then select proxy-tg.
   h. Health check type: select ELB.