# DNS Anycast with Cisco Prime Network Registrar

Anycast is a network and routing mechanism that enables a packet from a single client to go to one of many servers offering the same service. All the servers in the Anycast group are configured with the same Anycast IP address and the packet is routed from the client to the closest server by the best path as determined by routing algorithms. Anycast routing enables several important capabilities such as seamless redundancy, load balancing, and horizontal scaling by grouping multiple servers as a single service. Anycast DNS is simply an implementation of Anycast for DNS services. Anycast is used in conjunction with a routing protocol, such as Border Gateway Protocol (BGP) to advertise the availability of the service to an adjoining router, which makes the Anycast DNS to work effectively.

This chapter provides the knowledge and tools to configure Cisco Prime Network Registrar DNS services using Anycast.



# Basic Requirements for DNS Anycast

The following is a list of requirements and recommendations for supporting Anycast DNS:

- Clients should be configured to resolve DNS queries via the Caching DNS server's Anycast address(es).
- Nameservers should advertise their Anycast address in NS and A RRs.
- Nameservers should listen to DNS queries on the Anycast IP addresses.
- Nameservers should be configured with at least one Anycast IP address on a loopback interface.

- Additionally, the server should be configured with a management IP, which can be either a physical or an additional loopback interface.
- At least one physical IP must be defined on the DNS server for the exchange of routing information, as well as, system access and maintenance in the absence of the routes to the Anycast IP address(es).
- Nameservers should be configured to use the physical or management IP addresses for zone transfers, zone updates, and/or query source to ensure that these updates go to the intended server.
- Nameservers should Inject Anycast IP address(es) into the routed network using routing protocols such as RIP, OSPF, or BGP.

# Anycast Routing

Anycast can be manually configured, it is best implemented using routing protocols such as BGP or OSPF, which announces the Anycast destination address to its gateway router. Using a routing protocol to announce availability of the DNS service ensures that routers do not send DNS queries into a blackhole if the service goes down. As the Cisco Prime Network Registrar DNS application does not have routing capabilities, some code, external to the DNS application must be added to the DNS environment (physical server or virtual machine). The prominent and open source products are FRRouting (FRR) for RHEL/CentOS 8.x and Quagga for RHEL/CentOS 7.x.

## FRRouting

**Note** With RHEL/CentOS 8.x, use FRR.

FRR is an IP routing protocol suite for Linux and Unix platforms which includes protocol daemons for BGP, IS-IS, LDP, OSPF, PIM, and RIP.

FRR is forked from Quagga which is another routing protocol suite for Linux. FRR includes the fundamentals that made Quagga so popular as well as many enhancements that greatly improve on that foundation.

FRR does not ship with Cisco Prime Network Registrar. For more information about FRR, see the FRR documentation.

## Quagga

**Note** With RHEL/CentOS 7.x, use Quagga.

Quagga is a routing software suite, providing implementations of OSPFv2, OSPFv3, RIP v1 and v2, RIPng and BGP-4 for Unix platforms, Linux, Solaris, and NetBSD. The solution described in this chapter uses BGP.

The Quagga architecture consists of a core daemon, zebra, which acts as an abstraction layer to the underlying Linux kernel and presents the Zserv API over an Unix or TCP stream to Quagga clients. It is these Zserv clients, which typically implement a routing protocol and communicate routing updates to the zebra daemon.

Quagga daemons are configurable via a network accessible CLI (called **vty**). The CLI follows a style similar to that of other routing software. There is an additional tool included with Quagga called **vtysh**, which acts as a single cohesive front-end to all the daemons, allowing one to administer nearly all aspects of the various Quagga daemons in one place.

Quagga does not ship with Cisco Prime Network Registrar. For more information about Quagga, see the Quagga documentation.

# Script

A sample python script is included with Cisco Prime Network Registrar installation and is located at:

- FRR:

  **/opt/nwreg2/local/examples/dns/python/dns_anycast_bgp_frr.py**

- Quagga:

  **/opt/nwreg2/local/examples/dns/python/dns_anycast_bgp.py**

The script starts and stops FRR/Quagga, and monitors the DNS service by sending DNS queries to ensure that it is operational. When FRR/Quagga is started, its FRR/Quagga daemon sends an Anycast advertisement to the connected router making the DNS service available over the Anycast address. If the DNS server does not respond to queries from the script, the script will stop the FRR/Quagga daemon. Stopping FRR/Quagga breaks the TCP connection and the router will stop receiving BGP keep-alive messages. The router will then remove the DNS service from its Anycast group, and start sending DNS queries to the next closest and available DNS service. If the DNS server responds to queries from the script, the script checks to see if the FRR/Quagga daemons are running. If the daemons are not running, then the script starts the daemons.

It is recommended to copy the sample script to a different location, set up a cron job to periodically run the script to check the status of the DNS server (recommended interval is 5 minutes) and start or stop the BGP daemon accordingly. An example of a cron job is outside the scope of this solution.

# Router Configuration

Your configuration will probably be different based on your network requirements and variations in addressing schemes.

# Sample Anycast Configuration Using BGP

This section describes the basic setup and configuration of Anycast using BGP on a Cisco router and FRR/Quagga host-based routing software. The purpose of this section is not to instruct administrators on the configuration of routers and BGP, but to show a configuration that was successfully tested in the Cisco Prime Network Registrar labs. Note that your network requirements may be different.

BGP is a standardized exterior gateway protocol designed to exchange routing and reachability information among Autonomous Systems (AS) on the Internet. This configuration uses a single AS this recipe is not intended to be solution deployed across Autonomous Systems.

Perform the following steps on the hosts DNS-1 and DNS-2:

**For FRR:**

**Install FRR Routing Software**

Install FRR on the same system that is running Cisco Prime Network Registrar. This will install FRR package like the following:

**frr-7.0-5.el8.x86_64**

**For Quagga:**

**Install Quagga Routing Software**

Install Quagga on the same system that is running Cisco Prime Network Registrar. This will install Quagga package like the following:

**quagga-0.99.15-7.el6_3.2.x86_64**

**Create a Loopback Interface**

Create a loopback interface alias on the system. Configure the anycast IP address on this loopback interface.

On RHEL, the interface configuration files are located at **/etc/sysconfig/network-scripts**. Create a file in that directory named **ifcfg-lo:0** with the following contents:

```
DEVICE=lo:0
IPADDR=10.10.10.1
NETMASK=255.255.255.255
BOOTPROTO=none
ONBOOT=yes
```

Bring up the new loopback interface using the **ifup lo:0** command.

# Network Router Configuration

This router configuration is used in the validation of this DNS Anycast solution. It is provided as a reference to assist in the development of DNS Anycast solution. While it is a complete configuration for this specific solution, it is only intended to be a reference for developing your solution.

```
csr1000v# sh run
Building configuration...
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet1
 ip address 10.78.29.77 255.255.255.0 (Router)
 negotiation auto
!
interface GigabitEthernet2
 ip address 10.0.2.1 255.255.255.0 (Client)
 negotiation auto
!
interface GigabitEthernet4 (DNS-2)
 platform ring rx 256
 ip address 10.0.3.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet5 (DNS-3)
 platform ring rx 256
 ip address 10.0.5.1 255.255.255.0
```

```
 negotiation auto
!
router ospf 1
 router-id 2.2.2.2(is the loopback IP address)
 redistribute bgp 65500 subnets
 network 2.2.2.2 0.0.0.0 area 1
 network 10.0.6.0 0.0.0.255 area 1
 network 10.0.0.0 0.0.255.255 area 1
!
router bgp 65500
 bgp log-neighbor-changes
 neighbor IBGP peer-group
 neighbor IBGP update-source Loopback0
 neighbor ANY peer-group
 neighbor 192.0.2.1 remote-as 65500
 neighbor 192.0.2.1 peer-group IBGP
 neighbor 192.0.2.1 update-source Loopback0
 neighbor 10.0.3.2 remote-as65500
!(This should be the bgp AS in Quagga for DNS-2)
 neighbor 10.0.3.2 peer-group ANY
 neighbor 10.0.5.2 remote-as 65500
!(This should be the bgp AS in Quagga for DNS-3)
 neighbor 10.0.5.2 peer-group ANY
 !
 address-family ipv4
 redistribute ospf 1
 neighbor IBGP next-hop-self
 neighbor ANY next-hop-self
 neighbor 192.0.2.1 activate
 neighbor 10.0.3.2 activate
 neighbor 10.0.5.2 activate
 exit-address-family
!
virtual-service csr_mgmt
 ip shared host-interface GigabitEthernet1
 activate
!
ip default-gateway 10.78.28.1
ip forward-protocol nd
!
no ip http server
ip http secure-server
ip route 0.0.0.0 0.0.0.0 10.78.28.1
ip route 10.78.28.0 255.255.254.0 GigabitEthernet1 10.78.28.1
!
ip prefix-list anycast-ip seq 5 permit 10.10.10.1/32
!
control-plane
!
line con 0
 stopbits 1
line vty 0 4
 login local
!
!
end
```

# Configure FRRouting on DNS Servers

Configure the FRR configuration files on both the servers. The following example is for DNS-1. DNS-2 also needs to be configured similarly. The configuration files are located in **/etc/frr**.

There are number of example configuration files in **/etc/frr**: one for each routing protocol that FRR supports; one for zebra, the main process. For enabling Anycast using BGP, we need to configure **zebra.conf**, **bgpd.conf,** and **daemons** file.

# Enable zebra and bgpd in Daemons File

```
# cat /etc/frr/daemons
# This file tells the frr package which daemons to start.
watchfrr_enable=yes
watchfrr_options="-r '/usr/lib/frr/frr restart %s' -s '/usr/lib/frr/frr start %s' -k
'/usr/lib/frr/frr stop %s'"
#
zebra=yes
bgpd=yes
ospfd=no
```

# FRR Zebra Configuration

```
# cat /etc/frr/zebra.conf
hostname DNS-1
!
password zebra
enable password zebra
!
interface eth0
 ip address 10.0.3.2/24
!
interface lo
 ip address 10.10.10.1/32
!
line vty
!
```

**Note** Repeat the steps for any other Anycast servers that are part of the group.

# FRR BGP Configuration

```
# cat /etc/frr/bgpd.conf
! -*- bgp -*-
!
! BGPd sample configuration file
!
!
hostname DNS-1
password zebra
log stdout
!
router bgp 65500
bgp router-id 10.78.29.79
bgp log-neighbor-changes
network 10.10.10.1/32
timers bgp 4 16
neighbor 10.0.3.1 remote-as 65500
neighbor 10.0.3.1 next-hop-self
neighbor 10.0.3.1 prefix-list DEFAULT in
```

```
neighbor 10.0.3.1 prefix-list ANYCAST out
!
address-family ipv4
network 10.0.3.1/24
neighbor 10.0.3.1 activate
exit-address-family
!
ip prefix-list ANYCAST seq 5 permit 10.10.10.1/32
ip prefix-list DEFAULT seq 5 permit 0.0.0.0/0
line vty
!
```

## Start FRR Service

Start the FRR service using the following command:

```
systemctl start frr.service
```

### Create Additional IP address on the Loopback Interface

To create the additional IP address on the loopback interface for anycast using FRR, refer the Red Hat documentation.

## Restart FRR Service

Restart the FRR service using the following command:

```
systemctl restart frr.service
```

# Configure Quagga on DNS Servers

Configure the Quagga configuration files on both the servers. The following example is for DNS-1. DNS-2 also needs to be configured similarly. The configuration files are located in **/etc/Quagga**.

There are number of example configuration files in **/etc/Quagga**: one for each routing protocol that Quagga supports; one for zebra, the main process. For enabling Anycast using BGP, we need to configure **zebra.conf** and **bgpd.conf**.

## Quagga Zebra Configuration

```
# cat /etc/quagga/zebra.conf
hostname DNS-1
!
password zebra
enable password zebra
!
interface eth0
 ip address 10.0.3.2/24
!
interface lo
!
line vty
!
```

**Note** Repeat the steps for any other Anycast servers that are part of the group.

## Quagga BGP Configuration

```
# cat /etc/quagga/bgpd.conf
! -*- bgp -*-
!
! BGPd sample configuration file
!
!
hostname DNS-1
password zebra
log stdout
!
router bgp 65500
bgp router-id 10.78.29.79
bgp log-neighbor-changes
network 10.10.10.1/32
timers bgp 4 16
neighbor 10.0.3.1 remote-as 65500
neighbor 10.0.3.1 next-hop-self
neighbor 10.0.3.1 prefix-list DEFAULT in
neighbor 10.0.3.1 prefix-list ANYCAST out
!
address-family ipv4
network 10.0.3.1/24
neighbor 10.0.3.1 activate
exit-address-family
!
ip prefix-list ANYCAST seq 5 permit 10.10.10.1/32
ip prefix-list DEFAULT seq 5 permit 0.0.0.0/0
line vty
!
```

## Start BGP daemon

Start the BGP daemon using the following command:

```
systemctl start bgpd
```

# Run Diagnostics on Router

Run diagnostics on the router to make sure that the Anycast is set up properly.

The **sh ip bgp summary** command output shows that router-1 has opened a BGP session with the two neighbors. The value **State/PfxRcd** indicates that the TCP session is up and the routers and hosts are exchanging routes. This field should be a numeric value showing how many route prefixes have been received from the remote neighbor. The example value is 1. At this point, the BGP connection with the DNS servers is in Established state.

The **sh ip bgp** summary:

```
BGP router identifier 2.2.2.2, local AS number 65500
BGP table version is 86, main routing table version 86
```

```
1 network entries using 248 bytes of memory
2 path entries using 240 bytes of memory
1/1 BGP path/bestpath attribute entries using 248 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 736 total bytes of memory
BGP activity 16/15 prefixes, 61/59 paths, scan interval 60 secs
```

| Neighbor | V | AS | MsgRcvd | MsgSent | TblVer | InQ | OutQ | Up/Down | State/PfxRcd |
|---|---|---|---|---|---|---|---|---|---|
| 192.0.2.1 | 4 | 65500 | 0 | 0 | 1 | 0 | 0 | 4w0d | Idle |
| 10.0.3.2 | 4 | 65500 | 137919 | 129519 | 86 | 0 | 0 | 1w0d | 1 |
| 10.0.5.2 | 4 | 65500 | 137923 | 129519 | 86 | 0 | 0 | 1w0d | 1 |

The **show ip bgp neighbors** command shows information about the neighbors in detail.

The **show ip route** command should have an entry for the Anycast address and the host via which it is currently routed.

```
#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2
 i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
 ia - IS-IS inter area, * - candidate default, U - per-user static route
 o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
 a - application route
 + - replicated route, % - next hop override, p - overrides from PfR
B 10.10.10.1/32 [200/0] via 10.0.3.2, 00:00:10
```

# Monitor BGP Traffic Logs

To monitor the BGP traffic logs on the hosts DNS-1 and DNS-2, use the **telnet localhost bgpd** command.

**FRR:**

```
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is FRRouting (version 7.0).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
dns-anycast-1> enable
dns-anycast-1# terminal monitor
dns-anycast-1# conf t
dns-anycast-1(config)# debug bgp keepalives
dns-anycast-1(config)# 2020/10/27 02:56:22 BGP: : 10.0.3.1 KEEPALIVE rcvd

dns-anycast-1(config)# 2020/10/27 02:56:23 BGP: : 10.0.3.1 sending KEEPALIVE
2020/10/27 02:56:27 BGP: : 10.0.3.1 KEEPALIVE rcvd
2020/10/27 02:56:28 BGP: : 10.0.3.1 sending KEEPALIVE
```

```
2020/10/27 02:56:32 BGP: : 10.0.3.1 KEEPALIVE rcvd
2020/10/27 02:56:33 BGP: : 10.0.3.1 sending KEEPALIVE
2020/10/27 02:56:37 BGP: : 10.0.3.1 KEEPALIVE rcvd
2020/10/27 02:56:38 BGP: : 10.0.3.1 sending KEEPALIVE
2020/10/27 02:56:42 BGP: : 10.0.3.1 KEEPALIVE rcvd
2020/10/27 02:56:43 BGP: : 10.0.3.1 sending KEEPALIVE
2020/10/27 02:56:47 BGP: : 10.0.3.1 KEEPALIVE rcvd
2020/10/27 02:56:48 BGP: : 10.0.3.1 sending KEEPALIVE
```

**Quagga:**

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello, this is Quagga (version 0.99.15).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
User Access Verification
Password:
DNS-1> enable
DNS-1# terminal monitor
DNS-1# 2016/07/13 15:49:20 BGP: 10.0.5.1 send message type 4, length (incl. header) 19
2016/07/13 15:49:21 BGP: 10.0.5.1 rcv message type 4, length (excl. header) 0
2016/07/13 15:49:25 BGP: 10.0.5.1 send message type 4, length (incl. header) 19
2016/07/13 15:49:27 BGP: 10.0.5.1 rcv message type 4, length (excl. header) 0
```

# Configure DNS Zones

While this is the conclusion of setting up the Anycast functionality, administrators will need to complete the configuration of the DNS servers. See Managing Zones.

For more information, refer the following links:

- http://www.pacnog.org/pacnog6/IXP/Anycast-v10.pdf
- http://www.nongnu.org/Quagga
- https://frrouting.org/
- https://cumulusnetworks.com/learn/frrouting/
- https://bgpgeek.com/installing-frr/
- https://access.redhat.com/solutions/4967711
- https://access.redhat.com/solutions/4538371
- http://www.linuxjournal.com/magazine/ipv4-anycast-linux-and-Quagga
- http://ddiguru.com/blog/125-anycast-dns-part-5-using-bgp

**Note** The above links reference external websites and Cisco is not responsible for keeping them up-to-date. They are provided for reference only. If you find that the content is outdated or if you cannot access the links, please contact the website owner for updated information.