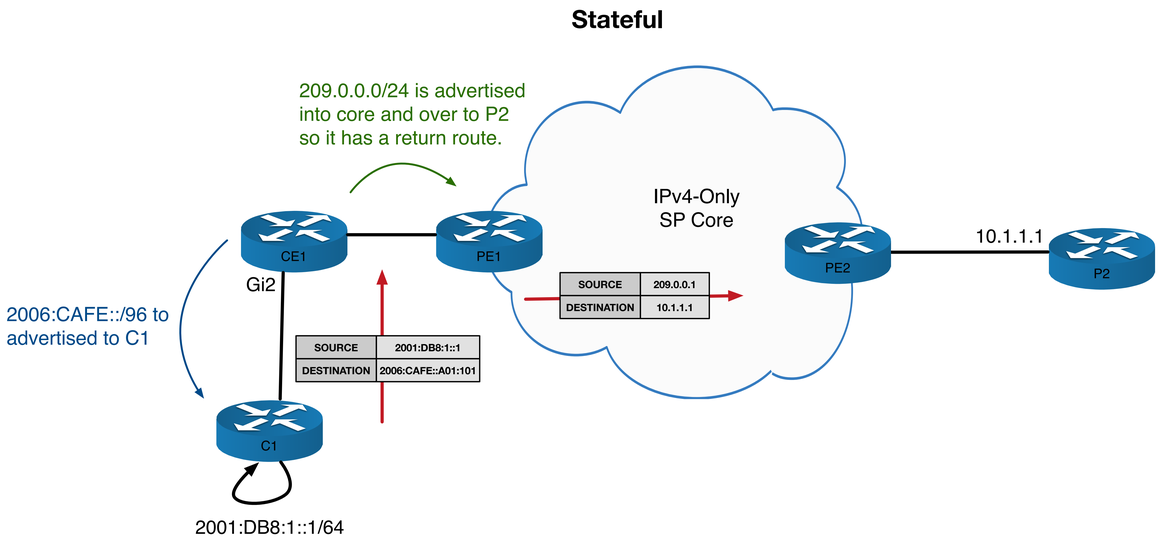
NAT64

NAT64 is all about changing IPv6 addresses to IPv4 addresses (both for the source and destinations). There are two way to approach this. Statefully, where the source translations are kept in a NAT table; and stateless, where the source IPv4 addresses are encoded into the source IPv6 addresses.

Stateful




```
hostname CE1
!
interface GigabitEthernet2
 nat64 enable
!
ipv6 access-list NAT64-ACL
 permit ipv6 2001:DB8:1::/64 any
!
nat64 prefix stateful 2006:CAFE::/96
nat64 v4 pool POOL1 209.0.0.1 209.0.0.254
nat64 v6v4 list NAT64-ACL pool POOL1 overload
```

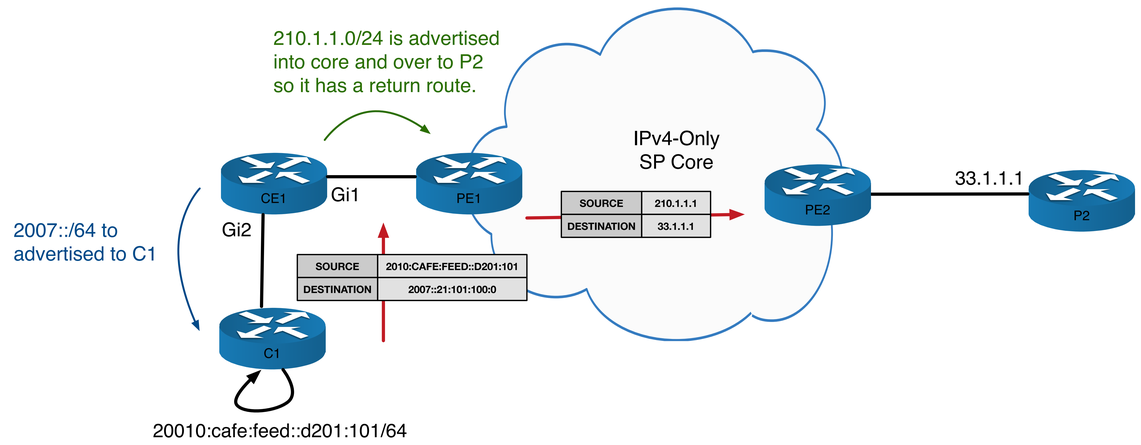
Any IPv6 traffic sourcing from **this ACL** will be source translated to **this IPv4 pool** and **PAT overloaded**. The destination is encoded within **this prefix**.

A static route is automatically created for **this IPv6 prefix** and **this IPv4 pool** - both of which will have a next hop of NVI0. These should both be advertised to their respective protocol networks to ensure reachability for remote hosts.

```
C1#ping vrf csc-cust ipv6 2006:CAFE::A01:101 source lo4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2006:CAFE::A01:101, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
C1#
```

```
CE1#sh nat64 translations

Proto  Original IPv4          Translated IPv4
       Translated IPv6        Original IPv6
---------------------------------------------------------------------------

icmp   10.1.1.1:2317          [2006:cafe::A01:101]:2317
       209.0.0.1:2317         [2001:db8:1::1]:2317

Total number of translations: 1

CE1#
```

Stateless




```
C1#ping 2007::21:101:100:0 source 2010:CAFE:FEED::D201:0101
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2007::21:101:100:0, timeout is 2 seconds:
Packet sent with a source address of 2010:CAFE:FEED::D201:101
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 90/97/102 ms
C1#
```

```
hostname CE1
!
interface GigabitEthernet2
 nat64 enable
 nat64 prefix stateless v6v4 2010:CAFE:FEED::/96
!
nat64 prefix stateless v4v6 2007::/64
!
nat64 route 210.1.1.0/24 GigabitEthernet2
```

```
CE1#sh nat64 translations

CE1#
```

There are not translations.
This is stateless.

Any IPv6 traffic whose source address fits into **this prefix** will be source translated into the corresponding IPv4 address in **this route command**. The prefix **must** have a complete IPv4 address encoded within it, otherwise the NAT will fail (remember state is not kept so everything must be provided in the IPv6 source/destination headers)

The IPv6 destination address will need to be witin **this prefix**. The corresponding IPv4 address will be extracted based on the prefix-length (in this case 64). How the IPv4 address is encoded based on the length is found in the table below.

A static route is automatically created for **this IPv6 prefix** and **this IPv4 route** - both of which will have a next hop of NVI0. These should both be advertised to their respective protocol networks to ensure reachability for remote hosts.

From RFC 6052

```
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|PL| 0-------------32--40--48--56--64--72--80--88--96--104---------|
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|32|     prefix    |v4(32)         | u | suffix                    |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|40|     prefix        |v4(24)     | u |(8)| suffix                |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|48|     prefix            |v4(16) | u | (16)  | suffix            |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|56|     prefix                |(8)| u |  v4(24)   | suffix        |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|64|     prefix                    | u |   v4(32)      | suffix    |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|96|     prefix                                    |    v4(32)     |
+--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

`2007::/64` has a /96 mask. So the encode for the source address seen in the above ping is as follows: