



DNS LOAD BALANCING

MASTER & SLAVE



NOVEMBER 4, 2020

DETAILS

Ipv4 Addr = 192.168.1.0/24

Master = spacex1.example.com

IP = 192.168.1.10

vNet = 4

Screenshot = M1....

Slave = spacex2.example.com

IP = 192.168.1.11

vNet = 4

Screenshot = S1....

Client = spacex3.example.com

IP = 192.168.1.20

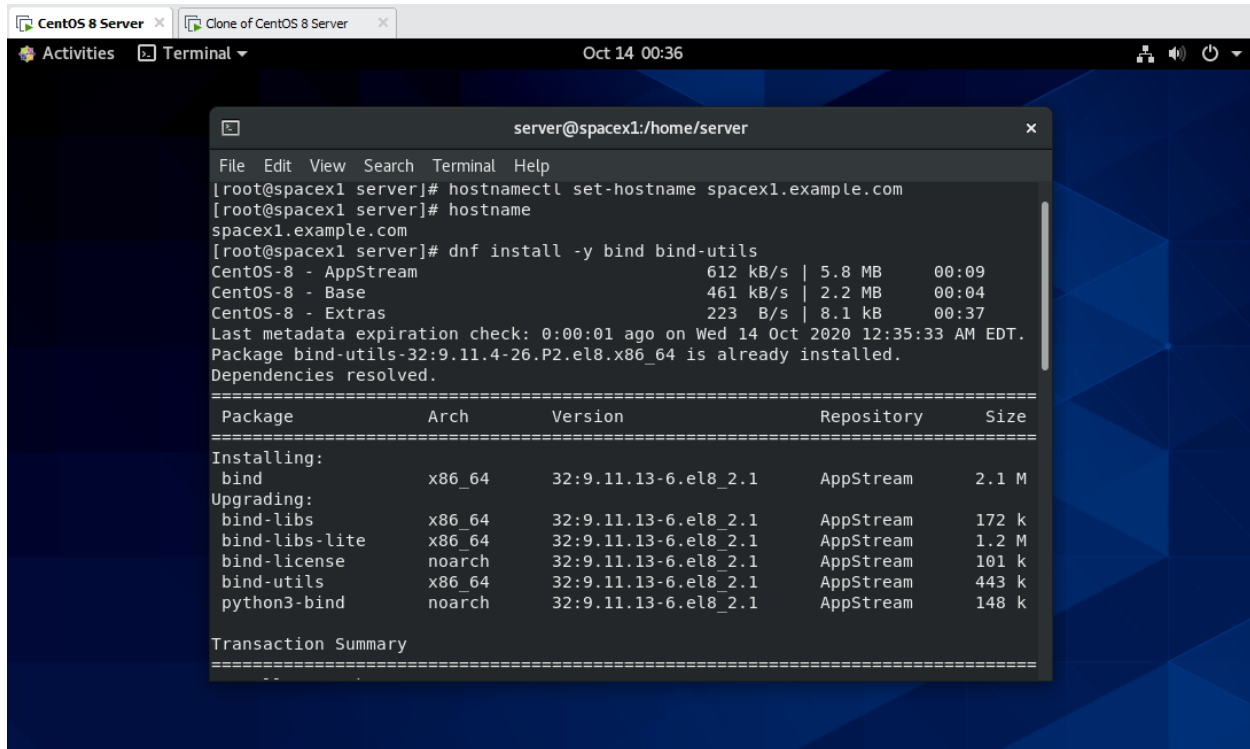
vNet = 4

Screenshot = C1....

FwZone = Forward Zone File

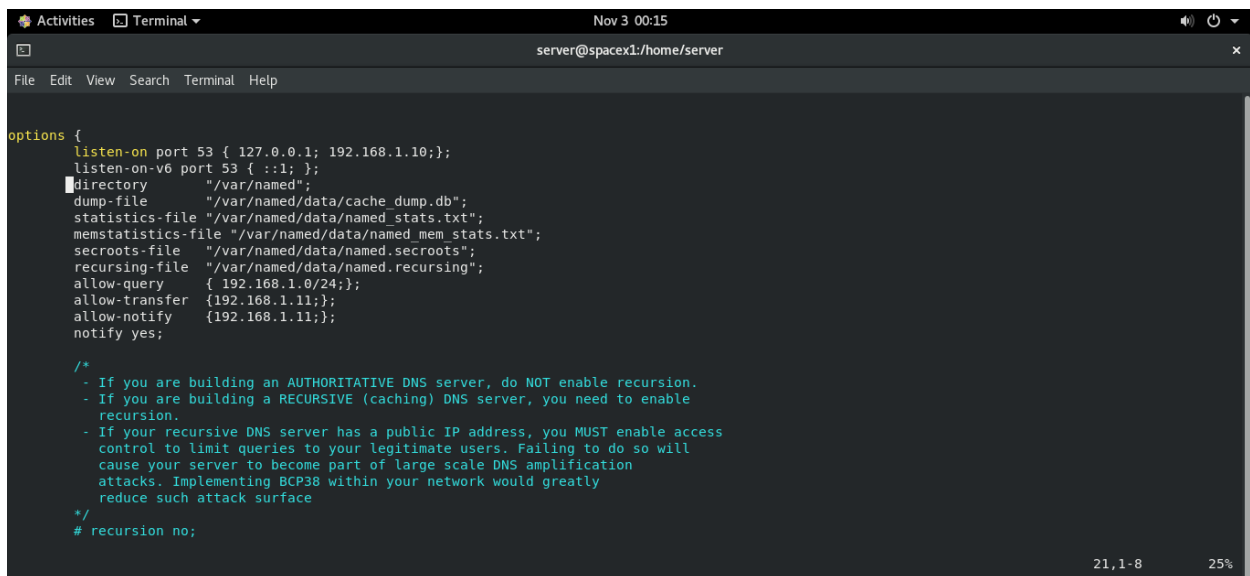
RvZone = Reverse Zone File

MASTER



A terminal window titled "server@spacex1:/home/server" showing the installation of bind and bind-utils. The user runs `hostnamectl set-hostname spacex1.example.com` and `hostname` returns `spacex1.example.com`. Then, `dnf install -y bind bind-utils` is executed. The output shows the progress of installing and upgrading packages. A table lists the packages being installed and upgraded, including their architecture, version, repository, and size. The transaction summary shows the total size of the packages to be installed.

```
server@spacex1:/home/server
File Edit View Search Terminal Help
[root@spacex1 server]# hostnamectl set-hostname spacex1.example.com
[root@spacex1 server]# hostname
spacex1.example.com
[root@spacex1 server]# dnf install -y bind bind-utils
CentOS-8 - AppStream                612 kB/s | 5.8 MB      00:09
CentOS-8 - Base                     461 kB/s | 2.2 MB      00:04
CentOS-8 - Extras                   223 B/s | 8.1 kB       00:37
Last metadata expiration check: 0:00:01 ago on Wed 14 Oct 2020 12:35:33 AM EDT.
Package bind-utils-32:9.11.4-26.P2.el8.x86_64 is already installed.
Dependencies resolved.
=====
Package                               Arch      Version                               Repository      Size
=====
Installing:
bind                                  x86_64    32:9.11.13-6.el8_2.1                 AppStream       2.1 M
Upgrading:
bind-libs                            x86_64    32:9.11.13-6.el8_2.1                 AppStream       172 k
bind-libs-lite                       x86_64    32:9.11.13-6.el8_2.1                 AppStream       1.2 M
bind-license                         noarch    32:9.11.13-6.el8_2.1                 AppStream       101 k
bind-utils                           x86_64    32:9.11.13-6.el8_2.1                 AppStream       443 k
python3-bind                         noarch    32:9.11.13-6.el8_2.1                 AppStream       148 k
Transaction Summary
=====
Install      1 Package
Upgrade     5 Packages
Total size: 3.7 MB
```



A terminal window titled "server@spacex1:/home/server" showing the configuration of the bind service. The user is editing the `/etc/named.conf` file. The configuration includes options for listening on port 53, directory, dump-file, statistics-file, memstatistics-file, secroots-file, recursing-file, allow-query, allow-transfer, allow-notify, and notify. There are also comments about enabling recursion and access control.

```
server@spacex1:/home/server
File Edit View Search Terminal Help
options {
    listen-on port 53 { 127.0.0.1; 192.168.1.10; };
    listen-on-v6 port 53 { ::1; };
    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";
    secroots-file "/var/named/data/named.secroots";
    recursing-file "/var/named/data/named.recursing";
    allow-query { 192.168.1.0/24; };
    allow-transfer { 192.168.1.11; };
    allow-notify { 192.168.1.11; };
    notify yes;

    /*
    - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
    - If you are building a RECURSIVE (caching) DNS server, you need to enable
      recursion.
    - If your recursive DNS server has a public IP address, you MUST enable access
      control to limit queries to your legitimate users. Failing to do so will
      cause your server to become part of large scale DNS amplification
      attacks. Implementing BCP38 within your network would greatly
      reduce such attack surface
    */
    # recursion no;
}
```


REVERSE ZONE

```
Activities Terminal Nov 3 01:28
server@spacex1:/var/named

File Edit View Search Terminal Help
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 86400
@ IN SOA spacex1.example.com. root.example.com. (
    2017020402 ; serial
    3600       ; refresh
    1800       ; retry
    604800     ; expire
    86400      ; minimum TTL
);
;nameservers
; IN NS spacex1.example.com.
; IN NS spacex2.example.com.
;
;nameserver IP addresses
; IN A 192.168.1.10
; IN A 192.168.1.11
;
; client IP Address
; IN A 192.168.1.20
;
;nameserver PTR records
10 IN PTR spacex1.example.com.
11 IN PTR spacex2.example.com.
;
; client PTR records
20 IN PTR spacex3.example.com.
~
~
~
"reverse.example.com" 25L, 604C 19,1 All
```

```
Activities Terminal Oct 14 01:51
server@spacex1:/home/server

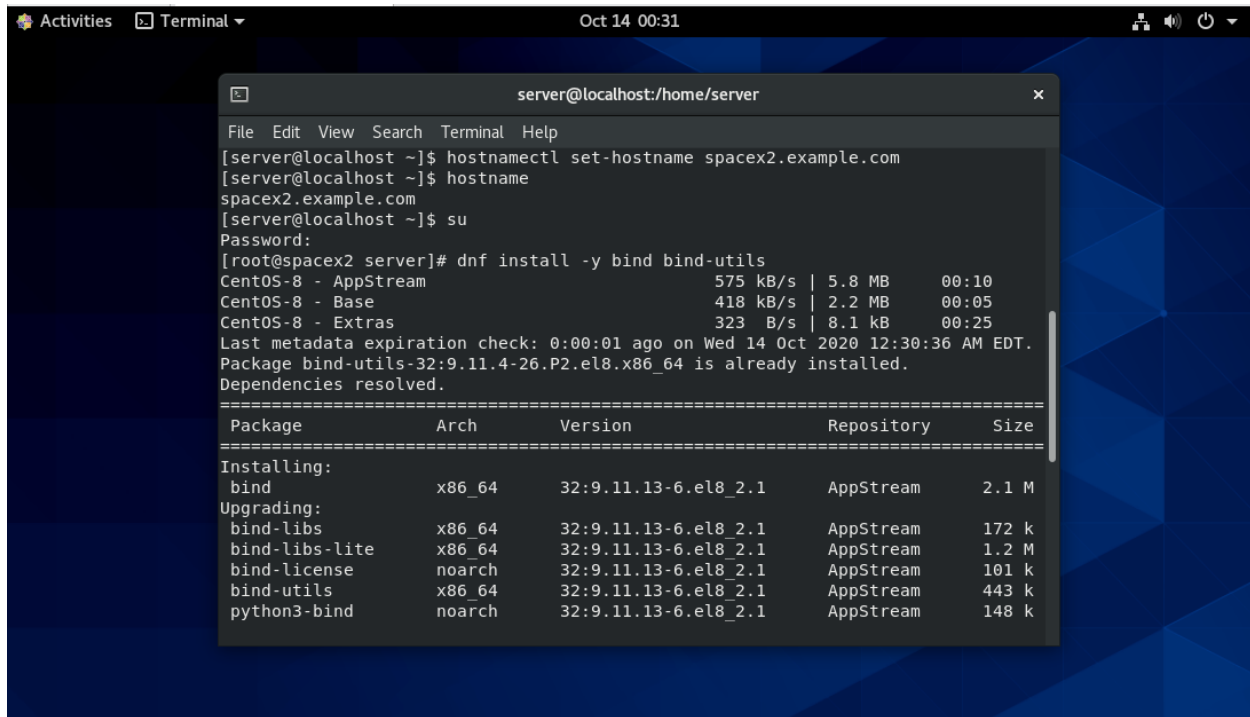
File Edit View Search Terminal Help
[root@spacex1 server]# nmcli con mod ens33 ipv4.gateway 192.168.1.1
[root@spacex1 server]# nmcli con mod ens33 ipv4.method manual
[root@spacex1 server]# nmcli con up ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkMa
nager/ActiveConnection/3)
[root@spacex1 server]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::b2ee:8440:9087:7850 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:03:5c:6d txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 38 bytes 5275 (5.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 808 (808.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 808 (808.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
Activities Terminal Oct 14 02:38
server@spacex1:/home/server
File Edit View Search Terminal Help
[root@spacex1 server]# named-checkconf
[root@spacex1 server]# named-checkzone example.com /var/named/forward.example.com
zone example.com/IN: loaded serial 2017020401
OK
[root@spacex1 server]# named-checkzone 1.168.192.in-addr.arpa /var/named/reverse.example.com
zone 1.168.192.in-addr.arpa/IN: loaded serial 2017020402
OK
[root@spacex1 server]# systemctl start named
[root@spacex1 server]# systemctl enable named
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/named.service.
[root@spacex1 server]# firewall-cmd --zone=public --add-port=53/tcp --permanent
success
[root@spacex1 server]# firewall-cmd --zone=public --add-port=53/udp --permanent
success
[root@spacex1 server]# firewall-cmd --reload
success
[root@spacex1 server]# firewall-cmd --add-service=dns --permanent;firewall-cmd --reload
success
success
[root@spacex1 server]#
```

```
Activities Terminal Nov 2 10:29
server@spacex1:/home/server
File Edit View Search Terminal Help
[root@spacex1 server]# hostnamectl
  Static hostname: spacex1.example.com
        Icon name: computer-vm
        Chassis: vm
        Machine ID: d830072eb7d842dca32a037d2097e998
        Boot ID: ec3699ca4123401ca6788a2673d489c6
        Virtualization: vmware
        Operating System: CentOS Linux 8 (Core)
        CPE OS Name: cpe:/o:centos:centos:8
        Kernel: Linux 4.18.0-147.el8.x86_64
        Architecture: x86_64
[root@spacex1 server]#
```

SLAVE

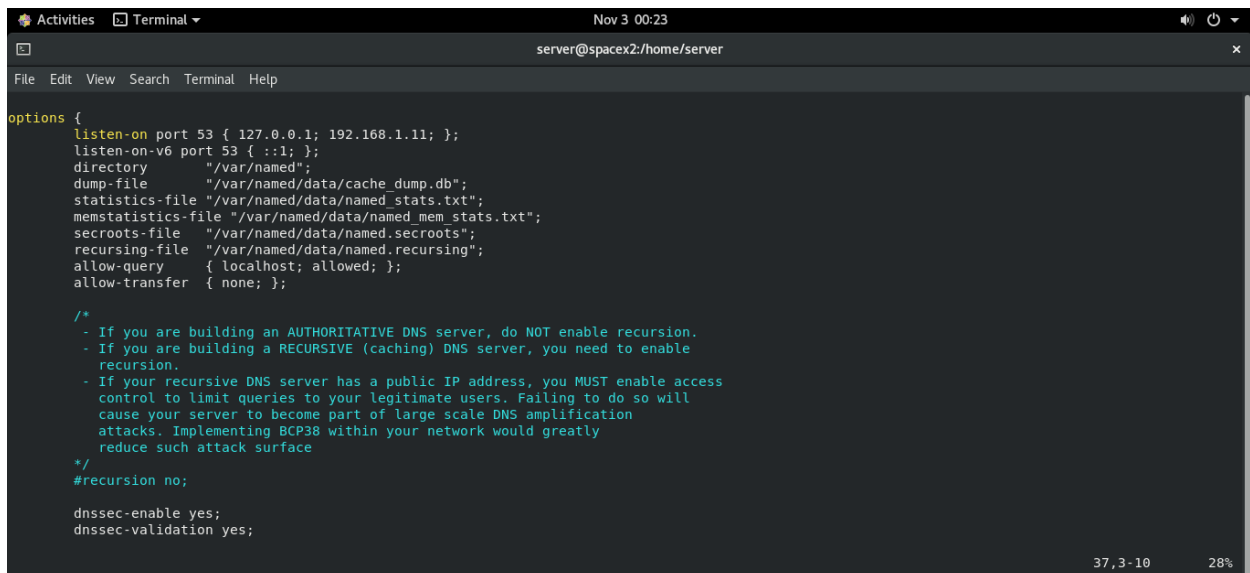


A terminal window titled "server@localhost:/home/server" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[server@localhost ~]$ hostnamectl set-hostname spacex2.example.com
[server@localhost ~]$ hostname
spacex2.example.com
[server@localhost ~]$ su
Password:
[root@spacex2 server]# dnf install -y bind bind-utils
```

The output of the dnf command shows the progress of installing bind and bind-utils. It includes a table of installed packages:

Package	Arch	Version	Repository	Size
Installing:				
bind	x86_64	32:9.11.13-6.el8_2.1	AppStream	2.1 M
Upgrading:				
bind-libs	x86_64	32:9.11.13-6.el8_2.1	AppStream	172 k
bind-libs-lite	x86_64	32:9.11.13-6.el8_2.1	AppStream	1.2 M
bind-license	noarch	32:9.11.13-6.el8_2.1	AppStream	101 k
bind-utils	x86_64	32:9.11.13-6.el8_2.1	AppStream	443 k
python3-bind	noarch	32:9.11.13-6.el8_2.1	AppStream	148 k



A terminal window titled "server@spacex2:/home/server" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the configuration of the bind service in /etc/named.conf:

```
options {
    listen-on port 53 { 127.0.0.1; 192.168.1.11; };
    listen-on-v6 port 53 { ::1; };
    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";
    secroots-file "/var/named/data/named.secroots";
    recursing-file "/var/named/data/named.recursing";
    allow-query { localhost; allowed; };
    allow-transfer { none; };
};

/*
- If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
- If you are building a RECURSIVE (caching) DNS server, you need to enable
  recursion.
- If your recursive DNS server has a public IP address, you MUST enable access
  control to limit queries to your legitimate users. Failing to do so will
  cause your server to become part of large scale DNS amplification
  attacks. Implementing BCP38 within your network would greatly
  reduce such attack surface
*/
#recursion no;

dnsssec-enable yes;
dnsssec-validation yes;
```

The terminal shows the configuration of the bind service in /etc/named.conf. The configuration includes options for listening on port 53, directory, dump-file, statistics-file, memstatistics-file, secroots-file, recursing-file, allow-query, allow-transfer, recursion, and dnsssec-enable. The recursion is set to no. The dnsssec-enable and dnsssec-validation are set to yes.

Activities Terminal Nov 3 00:23

server@spacex2:/home/server

```
File Edit View Search Terminal Help

logging {
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};

zone "." IN {
    type hint;
    file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";

# zone statement for forward dns lookup
zone "example.com" IN {
    type slave;
    file "slaves/forward.example.com";
    masters { 192.168.1.10; };
};

# zone statement for reverse dns lookup
zone "1.168.192.in-addr.arpa" IN {
    type slave;
    file "slaves/reverse.example.com";
    masters { 192.168.1.10; };
};
```

78,2 Bot

Activities Terminal Oct 14 03:02

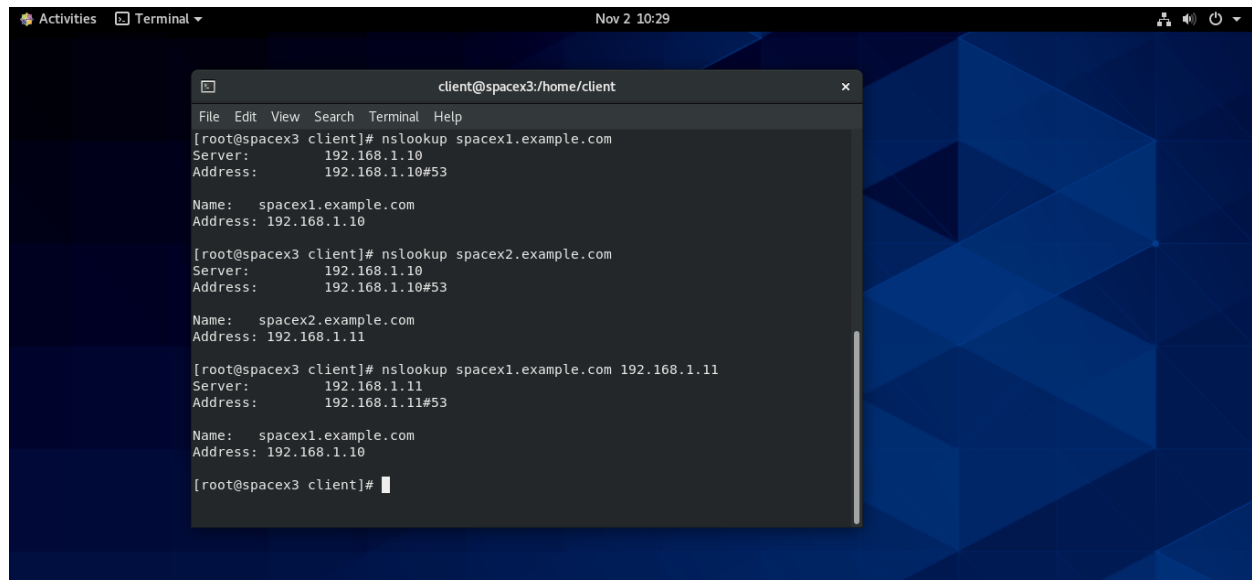
server@spacex2:/home/server

```
File Edit View Search Terminal Help

[root@spacex2 server]# nmcli con mod ens33 ipv4.addresses 192.168.1.11/24
[root@spacex2 server]# nmcli con mod ens33 ipv4.gateway 192.168.1.1
[root@spacex2 server]# nmcli con mod ens33 ipv4.method manual
[root@spacex2 server]# nmcli con up ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)
[root@spacex2 server]# nmcli con mod ens33 +ipv4.dns "192.168.1.10 192.168.1.11"
[root@spacex2 server]# nmcli con down ens33;nmcli con up ens33
Connection 'ens33' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
[root@spacex2 server]# firewall-cmd --zone=public --add-port=53/tcp --permanent
success
[root@spacex2 server]# firewall-cmd --zone=public --add-port=53/udp --permanent
success
[root@spacex2 server]# firewall-cmd --reload
success
[root@spacex2 server]# firewall-cmd --add-service=dns --permanent;firewall-cmd --reload
success
success
[root@spacex2 server]#
```




CONNECTION



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window has a title bar that reads "client@spacex3:/home/client". The terminal output shows three nslookup commands being executed. The first two commands are for "spacex1.example.com" and "spacex2.example.com", both returning the IP address 192.168.1.10. The third command is for "spacex1.example.com" with the IP address 192.168.1.11 specified, returning the IP address 192.168.1.11. The terminal window is set against a dark blue background with a geometric pattern.

```
client@spacex3:/home/client
File Edit View Search Terminal Help
[root@spacex3 client]# nslookup spacex1.example.com
Server:      192.168.1.10
Address:     192.168.1.10#53

Name:   spacex1.example.com
Address: 192.168.1.10

[root@spacex3 client]# nslookup spacex2.example.com
Server:      192.168.1.10
Address:     192.168.1.10#53

Name:   spacex2.example.com
Address: 192.168.1.11

[root@spacex3 client]# nslookup spacex1.example.com 192.168.1.11
Server:      192.168.1.11
Address:     192.168.1.11#53

Name:   spacex1.example.com
Address: 192.168.1.10

[root@spacex3 client]#
```