This article will show you how to setup and configure the BIND DNS Server. If you are looking for a guide on how to use DigitalOcean's integrated DNS service, you may want to review the ["How to Set Up a Host Name with DigitalOcean"](https://www.digitalocean.com/community/articles/how-to-set-up-a-host-name-with-digitalocean) article instead.  
  
Before we begin, it is recommended you have at least two cloud servers to run your nameservers. Two nameservers are suggested to assure your primary and secondary servers are redundant in case of failure. You may want to consider using two different POP's as well. For example, we've used San Francisco 1 and New York 1. For the purpose of this guide, it will be assumed you are configuring both a primary and secondary nameserver.  
  
It is worth noting that if you are managing a large number of domains this may not be the most viable solution, as you will need to manually add domains on both the master and slave nameservers. With that said, running your own nameservers is a great way to have more direct control over your hosting infrastructure, and assert full control over your DNS records.  
  
As with any new server, it's always important to ensure your system is up to date. You can verify this by checking for updates using *yum* as follows:

yum update -y

(Note: In DigitalOcean, we call our cloud servers as "droplets". We will use both terms throughout this tutorial)

Initial BIND Installation

To begin, we will need to install the BIND and BIND Utilities packages using *yum*.

yum install bind bind-utils -y

Next, we'll open the BIND (named) configuration file and make several modifications.

nano -w /etc/named.conf

Your "options" section should appear as follows, replacing 2.2.2.2 with the IP of your second droplet.

options {

#listen-on port 53 { 127.0.0.1; };

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";

allow-query { any; };

allow-transfer { localhost; 2.2.2.2; };

recursion no;

dnssec-enable yes;

dnssec-validation yes;

dnssec-lookaside auto;

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

};

Above, *listen-on* must be commented to listen on all available interfaces. Recursion should be turned off to prevent your server from being abused in "reflection" DDoS attacks. The*allow-transfer* directive whitelists transfers to your secondary droplet's IP. Furthermore, we have changed the *allow-query* directive to "any" in order to allow users proper access to hosted zones.  
  
Next, we'll want to add a new zone for our first domain, you should add the following to your named.conf below the existing zones.

zone "mydomain.com" IN {

type master;

file "mydomain.com.zone";

allow-update { none; };

};

After saving named.conf with the changes above, we're ready to create our first zone file.

Configure BIND Zones

Firstly, we'll need to open the zone file, using the name you specified in the configuration above. (Ex: mydomain.com.zone)

nano -w /var/named/mydomain.com.zone

We'll add the following contents to our newly created file. You should replace the applicable information with your own, where 1.1.1.1 is the IP of your first droplet, 2.2.2.2 is the IP of your second droplet and 3.3.3.3 is the IP you wish to point the domain itself to, such as a droplet running a webserver. You are free to add additional entries in the same format.

$TTL 86400

@ IN SOA ns1.mydomain.com. root.mydomain.com. (

2013042201 ;Serial

3600 ;Refresh

1800 ;Retry

604800 ;Expire

86400 ;Minimum TTL

)

; Specify our two nameservers

IN NS ns1.mydomain.com.

IN NS ns2.mydomain.com.

; Resolve nameserver hostnames to IP, replace with your two droplet IP addresses.

ns1 IN A 1.1.1.1

ns2 IN A 2.2.2.2

; Define hostname -> IP pairs which you wish to resolve

@ IN A 3.3.3.3

www IN A 3.3.3.3

We can now start named for the first time. This may take several minutes while named generates the *rndc.key* file, which only occurs on first execution.

service named restart

Once named has started successfully, we'll want to ensure that it is enabled as a startup service, by running the following:

chkconfig named on

By now, we should have a fully operational primary nameserver. You can verify that BIND is working correctly by running the following command, replacing 1.1.1.1 with the IP of your first droplet.

dig @1.1.1.1 mydomain.com

If you recieve a response which includes an answer and authority section, your nameserver has been configured correctly.

Slave Nameserver Configuration

With our primary nameserver configured, we'll now setup a slave nameserver on our second cloud server. As always, please assure your system is up to date by checking for updates with *yum* as follows:

yum update -y

We can start by installing BIND (and related utilities) on the second droplet, in the same manner as the first:

yum install bind bind-utils -y

We'll proceed by opening named.conf and making the same changes we made previously, ommitting the "*allow transfer*" line. This directive is unnecessary as we will only be transfering records from our primary nameserver.

nano -w /etc/named.conf

options {

#listen-on port 53 { 127.0.0.1; };

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";

allow-query { any; };

recursion no;

dnssec-enable yes;

dnssec-validation yes;

dnssec-lookaside auto;

/\* Path to ISC DLV key \*/

bindkeys-file "/etc/named.iscdlv.key";

managed-keys-directory "/var/named/dynamic";

};

We will add the zone we configured on the first droplet, this time changing the "type" directive to slave, instead of master. You should replace "1.1.1.1" with your first droplet's IP address.

zone "mydomain.com" IN {

type slave;

masters { 1.1.1.1; };

file "mydomain.com.zone";

};

After configuring our slave zone, we'll start named. Again this may take several minutes while our *rndc.key* file is initially generated.

service named start

As with the first cloud server, we want to assure named is set to run at startup with the following:

chkconfig named on

Your slave nameserver should now be up and running. You can verify that it is fully operational by using *dig* again, replacing 2.2.2.2 with the IP of your second droplet.

dig @2.2.2.2 mydomain.com

After any changes you make to the master zone files, you will need to instruct BIND to reload. Remember, you must also increment the "serial" directive to ensure synchronicity between the master and slave.  
  
To reload the zone files, we need to run the following command on the master nameserver, followed by the slave:

rndc reload

BIND in a chroot environment

It is generally advised to install the additional package "*bind-chroot*" which will drop the privileges of BIND into a chroot environment.  
  
Luckily, the CentOS package makes this extremely simple. The only aspect worth noting is that active paths for BIND will change to their chrooted equivalents, for example */var/named*becomes */var/named/chroot/var/named* With CentOS 6, you will not need to move any files as the package automatically creates hard symlinks to the non-chrooted directories.  
  
If you'd like to enable this feature for the added security which it provides, you can do the following:

yum install bind-chroot -y

service named restart