Oracle Linux 10

Configuring a BIND DNS Server

G24143-01
June 2025

Oracle Linux 10 Configuring a BIND DNS Server,

G24143-01

Contents

Preface

Oracle Linux 10: Configuring a BIND DNS Server provides information about using Berkeley Internet Name Domain (BIND) to set up a Domain Name System (DNS) on Oracle Linux 10 systems.

Documentation License

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also

mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About DNS and BIND

DNS is a network-based service that resolves domain names to IP addresses. For a small, isolated network you can use entries in the `/etc/hosts` file to provide the name-to-address mapping. However, most networks that are connected to the Internet use DNS.

DNS is a hierarchical and distributed database.

Consider the fully qualified domain name (FQDN) `wiki.us.example.com`. In this example, the top-level domain is `com`, `example` is a subdomain of `com`, `us` is a subdomain of `example`, and `wiki` is the host name.

Each of these domains are grouped into zones for administrative purposes. A DNS server, or *name server*, stores the information that's needed to resolve the component domains inside a zone. In addition, a zone's DNS server stores pointers to the other DNS servers that are responsible for resolving each subdomain.

If an external client requests its local name server to resolve a FQDN, such as `wiki.us.example.com` to an IP address for which that server isn't authoritative, the server queries a `root` name server for the address of a name server that's authoritative for the `.com` domain. This server then provides the IP address of another name server authoritative for the `example.com` domain, which in turn provides the IP address of the authoritative name server for `us.example.com`, and so on.

The querying process ends with the IP address for the FQDN being provided to the external client that made the request. This process is known as a recursive query, where the local name server handles each referral from an external name server to another name server on behalf of the resolver.

Iterative queries rely on the resolver being able to handle the referral from each external name server to trace the name server that's authoritative for the FQDN. Most resolvers use recursive queries and so can't use name servers that support only iterative queries.

Most Oracle Linux releases provide the BIND implementation of DNS. The `bind` package includes the DNS server daemon (`named`), tools for working with DNS, such as `rndc`, and some configuration files, including the following:

`/etc/named.conf`
Contains settings for `named` and lists the location and characteristics of the zone files for the domain. Zone files are typically stored in `/var/named`.

`/etc/named.rfc1912.zones`
Contains several zone sections for resolving local loopback names and addresses.

`/var/named/named.ca`
Contains a list of the root authoritative DNS servers.

2

Types of Name Servers

You can configure several types of name servers by using BIND, including the following:

Master name server
Authoritative for one or more domains, a primary (master) name server maintains its zone data in several database files, and can transfer this information periodically to any backup name servers that are also configured in the zone. An organization might maintain the following two primary name servers for a zone: one primary server outside the firewall to provide restricted information about the zone for publicly accessible hosts and services, and a hidden or *stealth* primary server inside the firewall that contains details of internal hosts and services.

Secondary or backup name server
Acting as a backup to a primary name server, a backup name server maintains a copy of the zone data, which it periodically refreshes from the primary server's copy.

Stub name server
A primary name server for a zone might also be configured as a stub name server that maintains information about the primary and backup name servers of child zones.

Caching-only name server
Performs queries on behalf of a client and stores the responses in a cache after returning the results to the client. This server isn't authoritative for any domains and the information that it records is limited to the results of queries that it has cached.

Forwarding name server
Forwards all queries to another name server and caches the results, which reduces local processing, external access, and network traffic.

In practice, a name server can be a combination of several of these types in complex configurations.

3
Installing and Configuring a Name Server

By default, you can use the BIND installation to configure a caching-only name server using the configuration settings that are provided in the `/etc/named.conf` file and other included files.
To configure a caching-only name server:

1. Install the bind package.

   ```
   sudo dnf install bind bind-utils
   ```

2. Edit `/etc/named.conf` and configure the settings required for the server.

   The configuration file included with the BIND installation enables only the localhost to query `named` and resolve IP addresses:

   ```
   options {
     listen-on port 53 { 127.0.0.1; };
     listen-on-v6 port 53 { ::1; };
     allow-query { localhost; };
     recursion yes;
   };
   ```

   For more information, see The named Configuration File.

   a. Specify the network interfaces on which `named` listens for queries.

      The following example configures `named` to listen on three interfaces on an IPv4 network: the localhost, a network interface with IP address `192.168.0.10`, and a network interface with the IP address `10.0.3.100`:

      ```
      listen-on port 53 { 127.0.0.1; 192.168.1.10; 10.0.3.100; };
      ```

   b. Specify the IP addresses of clients that are allowed to query this server.

      In the following example, the `localnets` keyword grants any clients on the same network as the server permission to make queries:

      ```
      allow-query { localhost; localnets; };
      ```

   c. Specify the IP addresses of clients that are allowed to access cached data.

      In the following example, IP addresses between `192.168.1.0` and `192.168.1.254` are allowed to access cached data.

      ```
      allow-query-cache { localhost; 192.168.1.0/24; };
      ```

   d. Specify the IP addresses of clients that are allowed to make recursive queries.

In the following example, IP addresses between `10.0.0.0` and `10.0.255.255` are allowed to receive recursively resolved data.

```
allow-recursion { localhost; 10.0.3.0/16; };
```

3. Save `/etc/named.conf`, then confirm that the syntax is correct.

```
sudo named-checkconf
```

4. If required, edit the zone files.

   For more information, see:

   - Resource Records in Zone Files
   - Resource Records for Reverse-Name Resolution

5. Configure the system firewall to accept incoming TCP connections to port 53 and incoming UDP datagrams on port 53:

```
sudo firewall-cmd --zone=zone --add-port=53/tcp --add-port=53/udp
```

   To make the change persist across reboots, include the `--permanent` option:

```
sudo firewall-cmd --permanent --zone=zone --add-port=53/tcp --add-
port=53/udp
```

   For more information about securing the firewall, see Oracle Linux 10: Configuring the Firewall.

6. Restart the `named` service and configure it to start following system reboots.

```
sudo systemctl enable --now named
```

4

DNS Configuration Files

Domains are grouped into zones that are configured through zone files. Zone files store information about domains in the DNS database. Each zone file contains directives and resource records. Optional directives apply settings to a zone or instruct a name server to perform certain tasks. Resource records specify zone parameters and define information about the systems or hosts in a zone.

Examples of BIND configuration files can be found in the `/usr/share/doc/bind/sample/etc` file.

The named Configuration File

The main configuration file for the `named` service is `/etc/named.conf`. Detailed configuration information is available in the `named.conf(5)` manual page and the BIND 9 Administrator Reference Manual.

Default Configuration

The following example comes from the default `/etc/named.conf` file that's installed with the `bind` package and which configures a caching-only name server:

```
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";
    secroots-file   "/var/named/data/named.secroots";
    recursing-file  "/var/named/data/named.recursing";
    allow-query { localnets; };
    recursion yes;

    dnssec-enable yes;
    dnssec-validation yes;

    /* Path to ISC DLV key */
    bindkeys-file "/etc/named.iscdlv.key";

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

   pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";

    /* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
    include "/etc/crypto-policies/back-ends/bind.config";

};
```

```
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";
```

Options Block

The `options` block defines the global server configuration options and sets defaults for other statements.

`listen-on`
Specifies the port on which `named` listens for queries.

`directory`
Specifies the default directory for zone files if a relative pathname is specified.

`dump-file`
Specifies where `named` dumps its cache if it crashes.

`statistics-file`
Specifies the output file for the `rndc stats` command.

`memstatistics-file`
Specifies the output file for `named` memory-usage statistics.

`allow-query`
Specifies which IP addresses might query the server. `localnets` specifies all locally attached networks.

`recursion`
Specifies whether the name server performs recursive queries.

`dnssec-enable`
Specifies whether to use secure DNS (DNSSEC).

`dnssec-validation`
Specifies whether the name server would validate replies from DNSSEC-enabled zones.

`dnssec-lookaside`
Specifies whether to enable DNSSEC Lookaside Validation (DLV) using the key in `/etc/named.iscdlv.key` defined by `bindkeys-file`.

Logging Block

The `logging` block activates the logging of messages to `/var/named/data/named.run`. The `severity` parameter controls the logging level, and the `dynamic` value means that this level can be controlled by using the `rndc trace` command.

Zone Block

The `zone` block specifies the initial set of root servers using a hint zone. This zone specifies that `named` consult `/var/named/named.ca` for the IP addresses of authoritative servers for the root domain (`.`).

Zone Definition Example

You can add definitions to the configuration file that are appropriate to the network environment. The following example defines settings for the service and the top-level definitions for zones:

```
include "/etc/rndc.key";

controls {
    inet 127.0.0.1 allow { localhost; } keys { "rndc-key"; }
};

zone "us.example.com" {
    type master;
    file "master-data";
    allow-update { key "rndc-key"; };
    notify yes;
};

zone "example.com" IN {
    type slave;
    file "sec/slave-data";
    allow-update { key "rndc-key"; };
    masters {10.1.32.1;};
};

zone "2.168.192.in-addr.arpa" IN {
    type master;
    file "reverse-192.168.2";
    allow-update { key "rndc-key"; };
    notify yes;
};
```

The `include` directive enables external files to be referenced so that sensitive data such as key hashes can be placed in a separate file with restricted permissions.

The `controls` block defines access information and the security requirements that are necessary to use the `rndc` command with the `named` server:

`inet`
Specifies which hosts can run `rndc` to control named. In this example, `rndc` must be run on the local host (`127.0.0.1`).

`keys`
Specifies the names of the keys that can be used. The example specifies using the key named `rndc-key`, which is defined in `/etc/rndc.key`. Keys authenticate various actions by `named` and are the primary method of controlling remote access and administration.

The `zone` blocks define the role of the server in different zones.

The following zone options are used:

`type`
Specifies that this system is the primary name server for the zone `us.example.com` and a backup server for `example.com`. `2.168.192.in-addr.arpa` is a reverse zone for resolving IP addresses to host names. See Resource Records for Reverse-Name Resolution.

`file`
Specifies the path to the zone file relative to `/var/named`. The zone file for `us.example.com` is stored in `/var/named/master-data` and the transferred zone data for `example.com` is cached in `/var/named/sec/slave-data`.

`allow-update`
Specifies that a shared key must exist on both the primary and backup name servers for a zone transfer to take place from the primary to the backup. The following is an example record for a key in the `/etc/rndc.key` file:

```
key "rndc-key" {
    algorithm hmac-md5;
    secret "XQX8NmM41+RfbbSdcqOejg==";
};
```

You can use the `rndc-confgen -a` command to generate a key file.

`notify`
Specifies whether to notify the backup name servers when the zone information is updated.

`masters`
Specifies the primary name server for a backup name server.

Resource Records in Zone Files

A resource record in a zone file contains the following fields, some of which are optional, depending on the record type:

Name
Domain name or IP address.

TTL (time to live)
The maximum time that a name server caches a record before it checks whether a newer one is available.

Class
Always `IN` for the Internet.

Type
Type of record, for example:

`A` (address)
IPv4 address corresponding to a host.

`AAAA` (address)
IPv6 address corresponding to a host.

`CNAME` (canonical name)
Alias name corresponding to a host name.

`MX` (mail exchange)
Destination for email addressed to the domain.

`NS` (name server)
Fully qualified domain name of an authoritative name server for a domain.

`PTR` (pointer)
Host name that corresponds to an IP address for address-to-name lookups (reverse-name resolution).

`SOA` (start of authority)
Authoritative information about a zone, such as the primary name server, the email address of the domain's administrator, and the domain's serial number. All records following a `SOA` record relate to the zone that it defines up to the next `SOA` record.

Data
Information that the record stores, such as an IP address in an `A` record, or a host name in a `CNAME` or `PTR` record.

The following example shows the contents of a typical zone file such as `/var/named/master-data`:

```
$TTL 86400        ; 1 day
@ IN SOA dns.us.example.com. root.us.example.com. (
            57 ; serial
            28800 ; refresh (8 hours)
            7200 ; retry (2 hours)
            2419200 ; expire (4 weeks)
            86400 ; minimum (1 day)
            )
              IN  NS      dns.us.example.com.

dns           IN  A       192.168.2.1
us.example.com  IN  A       192.168.2.1
svr01         IN  A       192.168.2.2
www           IN  CNAME   svr01
host01        IN  A       192.168.2.101
host02        IN  A       192.168.2.102
host03        IN  A       192.168.2.103
...
```

A comment on a line is preceded by a semicolon (`;`).

The `$TTL` directive defines the default time-to-live value for all resource records in the zone. Each resource record can define its own time-to-live value, which overrides the global setting.

The `SOA` record is mandatory and includes the following information:

`us.example.com`
The name of the domain.

`dns.us.example.com.`
The fully qualified domain name of the name server, including a trailing period (`.`) for the root domain.

`root.us.example.com.`
The email address of the domain administrator.

serial
A counter that, if incremented, tells `named` to reload the zone file.

refresh
The time after which a primary name server notifies backup name servers that they should refresh their database.

retry
If a refresh fails, the time that a backup name server should wait before attempting another refresh.

expire
The maximum elapsed time that a backup name server has to complete a refresh before its zone records are no longer considered authoritative and it will stop answering queries.

minimum
The minimum time for which other servers should cache information obtained from this zone.

An `NS` record declares an authoritative name server for the domain.

Each `A` record specifies the IP address that corresponds to a host name in the domain.

The `CNAME` record creates the alias `www` for `svr01`.

For more information, see the BIND 9 Administrator Reference Manual.

Resource Records for Reverse-Name Resolution

Forward resolution returns an IP address for a specified domain name. Reverse-name resolution returns a domain name for a specified IP address. DNS implements reverse-name resolution by using the special `in-addr.arpa` and `ip6.arpa` domains for IPv4 and IPv6.

The characteristics for a zone's `in-addr.arpa` or `ip6.arpa` domains are usually defined in `/etc/named.conf`, for example:

```
zone "2.168.192.in-addr.arpa" IN {
    type master;
    file "reverse-192.168.2";
    allow-update { key “rndc-key”; };
    notify yes;
};
```

The zone's name consists of `in-addr.arpa`, preceded by the network portion of the IP address for the domain, with its dotted quads written in reverse order.

If the network doesn't have a prefix length that's a multiple of 8, see RFC 2317 for the format that you need to use instead.

The `PTR` records in `in-addr.arpa` or `ip6.arpa` domains define host names that correspond to the host part of the IP address. The following example is take from the `/var/named/reverse-192.168.2` zone file:

```
$TTL 86400        ;
@ IN SOA dns.us.example.com. root.us.example.com. (
            57 ;
            28800 ;
            7200 ;
            2419200 ;
            86400 ;
            )
              IN  NS      dns.us.example.com.

1             IN  PTR     dns.us.example.com.
1             IN  PTR     us.example.com.
2             IN  PTR     svr01.us.example.com.
101           IN  PTR     host01.us.example.com.
102           IN  PTR     host02.us.example.com.
103           IN  PTR     host03.us.example.com.
...
```

For more information, see the BIND 9 Administrator Reference Manual.

5

Example Usage of the rndc Command

The `rndc` command enables you to administer the `named` service.

See the following manual pages for more information:

- `rndc(8)`
- `rndc-confgen(8)`
- `named(8)`

Enable Remote Usage of rndc

The `named` service is administered locally. If the service is configured in the `controls` section of the `/etc/named.conf` file, then you can also use the command line to manage `named` remotely. To prevent unauthorized access to the service, `rndc` must be configured to listen on the selected port (by default, port 953), and both `named` and `rndc` must have access to the same key. To generate a suitable key, use the `rndc-confgen` command:

```
sudo rndc-confgen -a
```

The command creates the `/etc/rndc.key` file.

Check the Status of the named Service

Check the status of the `named` service as follows:

```
sudo rndc status
```

```
number of zones: 3
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/1000
tcp clients: 0/100
server is up and running
```

Reload Configuration Files after Changes

If you change the `named` configuration file or zone files, the `rndc reload` command instructs `named` to reload the files:

```
sudo rndc reload
```

6

Example Usage of the host Command

The `host` utility is recommended for performing DNS lookups. Without any arguments, the command displays a summary of its command line arguments and options.

For more information, see the `host(1)` manual page.

Look Up IP Address

Look up the IP address for `host01`:

```
host host01
```

Perform a Reverse Lookup

Perform a reverse lookup for the domain name that corresponds to an IP address:

```
sudo host 192.168.2.101
```

Query DNS for an IP Address

Query DNS for the IP address that corresponds to a domain:

```
sudo host dns.us.example.com
```

Display Verbose Output

Display verbose information about records of a certain type by using the `-v` and `-t` options:

```
sudo host -v -t MX  www.example.com


Trying "www.example.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49643
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.com.            IN    MX

;; ANSWER SECTION:
www.example.com.      135    IN    CNAME    www.example.com.
www.example.com.      1240      IN    CNAME    d4077.c.example.com.

;; AUTHORITY SECTION:
c.example.com.    2000    IN    SOA    m0e.example.com.
hostmaster.example.com. ...

Received 163 bytes from 10.0.0.1#53 in 40 ms
```

The `-a` option, which is equivalent to the `-v`, `-t`, and `ANY` options displays all of the available records for a zone. For example:

```
sudo host -a www.us.example.com


Trying "www.us.example.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40030
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.us.example.com.            IN    ANY

;; ANSWER SECTION:
www.us.example.com.        263    IN    CNAME    www.us.example.com.

Received 72 bytes from 10.0.0.1#53 in 32 ms
```