Primary and secondary server setup for 4PSA DNS Manager

This article describes how to configure two 4PSA DNS Manager servers in a master-slave setup that provides DNS services redundancy.

The described setup allows all the zones hosted on the primary 4PSA DNS Manager server to be automatically exported to the secondary 4PSA DNS Manager server as slave zones. The secondary server acts as a backup DNS server for the zones imported from the primary server. If the primary DNS server becomes unavailable, the secondary one will handle the zones' requests.

The present setup describes a one way replication. The clients are allowed to log in and make changes solely on the primary 4PSA DNS Manager server, while the secondary server only replicates these changes automatically. In case of a failure on the primary server, the clients will not be able to log in or make any changes to their zones until the primary server is restored. It is also possible to have multiple slave servers that synchronize from the master.

Requirements

Before starting the replication, check if the following requirements are met:

- Make sure that both servers employed in the configuration are updated to the latest 4PSA DNS Manager version.
- If you are using a firewall, then you must allow connections from the secondary server's IP to port 443 on the primary server.
- Ports 53 and 953 (UDP and TCP) must be opened on both servers.

Configuration

After assuring that all the requirements are met, you can start to configure both the primary and secondary servers.

Primary server

STEP 1: Log in to your 4PSA DNS Manager server using your favorite SSH console (e.g. `Putty`).

STEP 2: Go to the following directory:

> *DNSMANAGER_ROOT_D/remote/dnsmanager/*

where `DNSMANAGER_ROOT_D` is usually `/usr/local/dnsmanager`.

STEP 3: In this directory you will find a shell script called `dnsmanager_export.sh` that exports all the zones from a 4PSA DNS Manager server to a text file that can be later imported by another server.

You need to modify the script so that it converts all the exported zones to slave zones and it saves the output file in a directory where it can be accessed by the secondary server using an URL.

Because at the next 4PSA DNS Manager upgrade this script will be overwritten with the default one, you should make a copy of the script and edit the copy. This will allow you to use the copy of the script in the cron job regardless how many 4PSA DNS Manager upgrades were installed.

STEP 4: Edit the script and set the following variables:

`dump_file`

This variable defines the path of the file containing the zones. You must set it to `/usr/local/dnsmanager/admin/htdocs/zones_dump.txt` by running:

```
dump_file="/usr/local/dnsmanager/admin/htdocs/zones_dump.txt"
```

`dump_temp_file`

This variable defines the temporary directory for the dump file:

```
dump_temp_file="dump_file-$$"
```

`dump`

This variable sets the type of the zones that will be dumped (master/slave/both). You must set it to `both` by running:

```
dump="both"
```

`master2slaves`

This variable converts the master zones to slave zones. You must set it to `yes` using the following command:

```
master2slaves="yes"
```

STEP 5: Set up a cron job running the script to export the zones at regular time intervals to the location of your choice. The secondary server will be able to access the exported file at the following URL:

```
https://master_server_ip/zones_dump.txt
```

where `master_server_ip` is the IP address of the primary 4PSA DNS Manager server.

Example of a crontab line used to export the records every 10 minutes:

```
*/10 * * * * /usr/local/dnsmanager/remote/dnsmanager/dnsmanager_export.sh >/dev/null 2>&1
```

STEP 6: In the interface, add the secondary 4PSA DNS Manager server's IP to the **Slave DNS server IP or IP/Mask address** field from the **Settings > Global transfer IPs** page to allow zone transfers to the secondary 4PSA DNS Manager server.

Secondary server

On the secondary 4PSA DNS Manager server, you need to set up one client to host the slave zones which will be imported from the primary 4PSA DNS Manager server. Name the client `Secondary`. After the client has been created, follow the next steps:

STEP 1: Go to the **Clients Management** page and click the name link of the previously defined `Secondary` client account.

STEP 2: Next, click the **Remote Updates** icon available in the **Tools** section.

STEP 3: Here you have to set up the connection as follows:

- Fill in `https://master_server_ip/zones_dump.txt` for the **Remote update location** option. `master_server_ip` is the IP of the primary 4PSA DNS Manager server.
- In the **Add the following master IP** field, enter the IP address of the primary 4PSA DNS Manager server.

For the other available options, check the 4PSA DNS Manager help.

STEP 4: Go to the **Clients Management** and click on the *Secondary* client account, next go to **Client Settings** and check that the client has permissions to Manage zones and records. The client should have permission to manage all types of zones, otherwise it won't be able to import them.

The zone updates from the primary server will be handled by the `updateurld` that runs by default on every 4PSA DNS Manager installation.

Make sure that the client you added the **Remote update location** for has full permissions over zones management and the limits imposed on his account are not restrictive (for example, if you have 200 zones in the remote update location and the client zones limit is 100, only one hundred zones will be imported, which is less than desired).

Related articles

- Primary and secondary server setup for 4PSA DNS Manager
- How to block specific countries from accessing your server
- How to find out how many DNS queries are being made
- How to dump zones remotely from a Plesk Windows server
- How to debug Asterisk and Kamailio