Configuring Secure64 x86

Configuring Secure64 x86 Support

- Configuring Secure64 x86 Authority Server

A note on Ports

ProVision uses port 22 to communicate with and configure Secure64 infrastructure - please ensure that this is addressed in any ACLs /firewalls

ProVision also uses port 53 to do zone checks if the DNS Module is enabled and in use. Please ensure that your Secure64 infrastructure is configured to accept DNS lookups from the ProVision server

The initial setup of the Secure64 x86 Authoritative server is as follows:

Configuring Secure64 x86 Authority Server

1. Using the terminal, SSH to the desired S64 x86 server. All the configurations and zones will be pushed in /srv/knot/6c/
2. Create new user "provision"

```
sudo useradd provision
sudo passwd provision
```

3. Create a directory called "6c" that will be used for the ProVision exports and set the permissions, so that we can push the configurations in /srv/knot/

```
sudo mkdir --mode=u+rwx,g+wrs,o-rwx /srv/knot/6c
sudo chown provision.knot /srv/knot/6c
```

4. In order to properly reload the server without asking for a password from ProVision, we must edit the sudoers file to let the user "provision" execute "sudo knotc reload" after push. So, add the following line at the end of /etc/sudoers:

```
provision ALL=(ALL) NOPASSWD: /sbin/knotc reload
```

5. Add knot to be inside the "provision" group by editing the /etc/group as follows:

```
provision:x:120:knot
```

6. Add provision to be a part of the knot group (in case knot replaces a zone), so that Provision is able to replace it again on push. So, we must once again edit /etc/group, as follows:

```
knot:x:119:provision
```

7. Configure ProVision in the S64 server and add the include directive in /etc/knot/knot.conf:

```
include: /srv/knot/6c/6c_knot.conf
```

8. Open the ProVision UI and navigate to the DNS section to add the S64x86 server. While adding server settings, you must verify that Post Command is set to “sudo knotc reload” and the configuration path to “/srv/knot/6c/6c_knot.conf”

9. An example server configuration for the S64 x86 server is shown below:

Server Settings :

Edit the comment.

Common Settings

Display Name:

Example Secure64 x86 Server

This is the server name that will appear in the DNS interface.

FQDN or IP:

1.2.3.4

The IP address that ProVision will use to connect to this server.

DNS Port:

default: 53

The port will be used for DDNS and DNS Queries to the server.

Server Type:

Master

Export Zones: ON

DNS Service:

Secure64 x86 Authority

Parent Resource:

TLR

The new server resource will be a child of the Parent Resource.

Enable Records Check: ON

ProVision is checking if the DNS responds with a proper values to the zone records. In order record monitoring to work properly, you must enter a proper NameServer as "FQDN or IP" field.

Enable TSIG Key for transfers: OFF

If enabled, the provided TSIG key will be added to the ACL config of the related DNS Servers.

Secure64 x86 Settings

SSH Public Key Authentication: OFF

Please choose your SSH authentication type.

Username:

provision

Username for the SSH connection. It must have write access to the Bind configurations and zone folders.

Password:

••••

Port:

22

Server SSH Port.

SSH Route:

Use ssh routes in order to define a chain from ssh jump hosts.

Remote Directory:

/srv/knot/6c/

Path to the remote server where to store the generated zone files.

Knot Conf Path:

/srv/knot/6c/6c_knot.conf

Path to the 6connect knot.conf config.

Public IP Address:

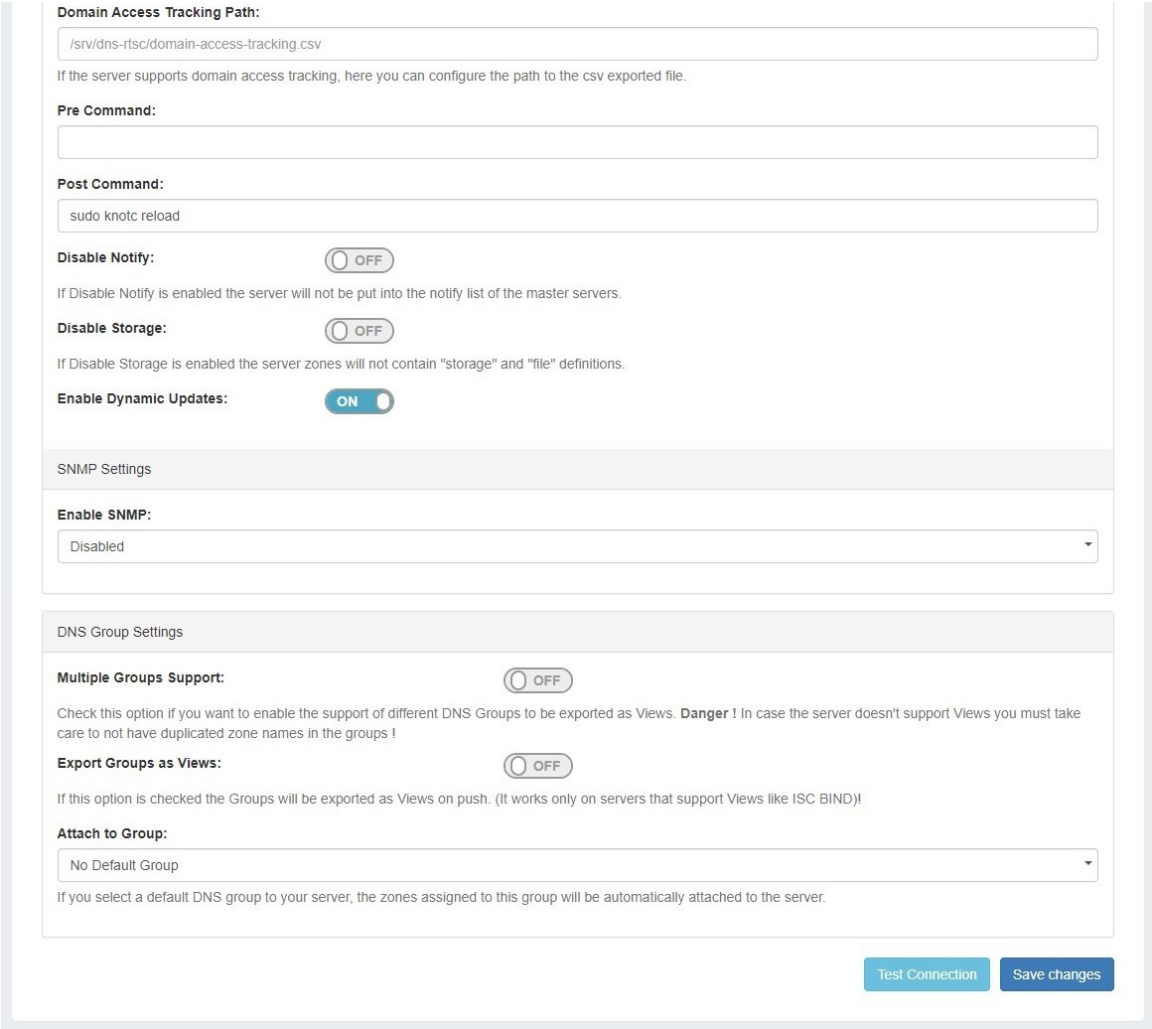
1.2.3.4

The public-facing IP address for this server used in writing server configs. If omitted, the 'FQDN or IP' field will be used.

Via IP Address:

123.123.123.123

If provided the value will be placed into the "via:" configuration on the remote section.



OTHER Record Types

When working with DNS Zones and Records, additional record types may be manually added by selecting "Other" when adding a new record.

S64 DNS users can use record type "Other" to add "SYNTH" or "TYPE65464" type records similar to the format below:

```
$ORIGIN 30 IN TYPE65464 ${p4} PTR ${a4}.pool.example.com.
$ORIGIN 600 IN TYPE65464 ${a4} A ${a4}
$ORIGIN TYPE65464 ${p6} PTR user${a6}.my.example.com.
$ORIGIN 5 IN SYNTH user${a6} AAAA ${a6}
$ORIGIN IN SYNTH nptr-${u} NAPTR 10 20 "A" "" "" srv-${u}
$ORIGIN IN SYNTH srv-${u} SRV 10 20 1234 srv-addr-${u}
```

However, arbitrary / other record types are unable to be validated, so use with care!