Migrate ISPConfig3

Disclaimer: Daily Data is a subscriber to ISPConfig/Howto Forge. That is the only association between us, and anything written here is the opinion and procedure interpreted by Daily Data. For the most accurate information, visit https://www.ispconfig.org/ and https://www.howtoforge.com/. If our procedures conflict with theirs, they are the authority.

Warning: It is not clear from the documentation, but your web services will be down during the migration. It is possible there are other services which will suffer downtime. Be prepared to do this at a time when your web server can be down. After successful completion, the sites become available again, but some failures may result in this step not being taken. See the section on Recovery at the bottom if this occurs.

Notice: A large majority of the time is taken setting proper permissions on the target server. Thus, the number of files in your users web sites and the speed of your hard disk will be the main factors in how long the migration takes.

Determine Overall Procedure

First, decide which is more valuable, your time or your money. If you have a lot of free time and want to know the ins and outs, you can do a manual install. A manual move requires you copy all content, configuration and databases to the new server.

The simplest way to migrate one ISPConfig installation to another is to purchase the Migration Toolkit from https://www.ispconfig.org/add-ons/ispconfig-migration-tool/. The developers have put a lot of time into making this simple, and are constantly upgrading the tool. I estimated it would take 4-8 hours to do a manual move, with client downtime of half that. With the migration tool, I was able to move three servers with a client downtime of **PUT HOURS USED HERE**.

The following documents moving a multiserver ISPConfig3 installation to a new set of target servers. All servers are virtuals.

NOTE: On another system, I did an operating system upgrade (Debian Wheezy to Devuan ASCII), then upgraded ISPConfig. The result was fairly inconsistent, mainly because of the massive jump of the operating system (two major releases) and took a significant time to tune after the move.

The move is from *Source* to *Target*, both defined below.

Source

- Debian Wheezy
- ISPConfig v3.1.6
- Multi Server
    - Web/Database/Control
    - Mail
    - DNS

Target

- Devuan ASCII (equiv to Debian Stretch)
- ISPConfig v3.1.14p2
- Multi Server
    - Web/Database/Control
    - Mail
    - DNS

I have added a LAN IP address for all the source servers (security alert!), and have set up the new servers with LAN IP's also. I will do the migration over the LAN. At the end of the procedure, I will shut down the Source servers, then change the IP's on the Target servers as per https://www.howtoforge.com/community/threads/changing-ip-address-of-ispconfig-3-installation.52236/

Install ISPConfig on new servers

The HowTo at https://www.howtoforge.com/tutorial/perfect-server-debian-9-stretch-apache-bind-dovecot-ispconfig-3-1/ is mainly for a single server. To install on a multi-server environment, you may want to purchase the manual https://www.ispconfig.org/documentation/user-manual/ since it defines which services are needed on which servers and the cost is cheap. It is, however, set up for Debian/Devuan Jessie, so some minor changes (mainly installing PHP, version 7 vs 5.2) are needed. The authors of both the web site and the manual have gone to great lengths to be clear on how to do the installation.

Note, we use Devuan, without systemD. If you also do this, you will need to modify some of the commands in the HowTo since they were written for Debian.

Warning: There is an error in the current manual. On the mail and dns servers, you also need to run:

```
apt-get install php7.0-mbstring
```

Set up migration tool

This is mainly taken from https://www.howtoforge.com/tutorial/how-to-migrate-ispconfig-confixx-plesk-to-ispconfig-31-single-server/, modified for multi-server setup.

- Make a copy of the database dbispconfig on the *Source* server.

```
mysqldump -u root -p dbispconfig > good_ispconfig.dmp
```

- Verify mysql on the *Target* server has **max_allowed_packet = 128M**. Be sure to restart mysql (not reload, restart).

```
find /etc/mysql -type f -exec sed -i
's/max_allowed_packet.*/max_allowed_packet = 128M/' \{\} \;
service mysql restart
```

- Create Remote User on *Target* admin server via ispConfig web ui

    - System | Remote Users | Add new user
    - username 'remoter' or something and add a password
    - check ALL the functions
    - Save user
- Create passwordless connection for root between Source and Target, for each server pair (web, mail, dns). See ssh Quick Reference if you need help.
- Install php-soap and rsync on all *Target*'s

```
apt-get -y install php-soap rsync
```

- Download migration toolkit (https://www.ispconfig.org/downloads/ispconfig_migration_toolkit-latest.zip) onto each *Source* server. Unzip it, then go into the created directory and un-tar/gzip the migration tool (ignore the copy tool for this since it is for a different purpose). Following is directly stolen from https://www.howtoforge.com/tutorial/how-to-migrate-ispconfig-confixx-plesk-to-ispconfig-31-single-server/

```
cd /root
mkdir migration
cd migration
wget https://www.ispconfig.org/downloads/ispconfig_migration_toolkit-latest.zip
unzip ispconfig_migration_toolkit-latest.zip
mv ispconfig_migration_toolkit/ispc3-migration-tool-*.tar.gz /root/migration/migtool.tar.gz
tar xzf migtool.tar.gz
rm -f migtool.tar.gz
```

You are now ready to begin the migration.

Perform Migration

For me, I had a backup of both the web and mail servers, so I first did an rsync of the backup to each machine. That allowed me to seed the migration, and reduce load on running web/mail servers while the migration was going on. The migration can be run multiple times and will simply add new files and update existing ones (but will not delete anything, so your mail grows and grows, but more on that later).

From my backup server, I ran the following.

```
rsync -av --delete --bwlimit=12800 /path/to/backup/of/web/clients/* newwebserver:/var/www/clients/
rsync -av --delete --bwlimit=12800 --exclude '/mailfilters/' --exclude '/postqueue/' --exclude '/quota.group' --exclude '/quota.user' --exclude 'sieve/' /path/to/backup/of/vmail/* newwebserver:/var/vmail/
```

This takes a long time, but can be run multiple times, and the bwlimit decreases the load on the LAN. On older versions, it did not take a modifier, so I put it in as KB/s, so this is 1Mb/s.

Note: Before you run the migration tool, you should determine a temporary directory to be used. This should be large enough to store the single biggest database you have, most likely. If you are unsure, dump the largest database you have and see how big the temp directory needs to be.

Now, run the migration tool from the *Source* primary server. The one with the ISPConfig3 interface on it. The first time you do this, it should be a dry run. It will ask you for a bunch of information and create a configuration file so subsequent runs can be made easily.

```
./migrate --source-temp-dir=/home/temp --target-temp-dir=/home/temp
```

We are passing two additional parameters since, in the case of our machines, we do not have enough space in /tmp to save some of the larger files like mysqldump outputs. However, I created a directory in /home, which has enough for the largest database dump that will happen.

testoutput

```
*** Disclaimer of Warranties ***
ISPConfig disclaims to the fullest extent authorized by law any and all
other warranties, whether express or implied,
including, without limitation, any implied warranties of title, non-
infringement, integration, merchantability or
fitness for a particular purpose.
By continuing to use this software, you agree to this.

First of all the most important question:
Do you want to make a real migration or a dry run? (dry, migrate): dry
The following modules are available:
        1. Confixx to ISPConfig 3 migrator (confixx)
        2. ISPConfig 3 to ISPConfig 3 migrator (ispconfig3)
        3. Plesk (10.5 - 17.5) to ISPConfig 3 migrator (plesk)
        4. ISPConfig 2 to 3 migrator (ispconfig2)
        5. AutoDNS to ISPConfig 3 migrator (autodns)

Which one is the one matching your SOURCE system? (confixx, ispconfig3,
plesk, ispconfig2, autodns): ispconfig3
Enter this (SOURCE) server's ip that is used for outgoing connections
[10.27.45.1]:
Testing MySQL connection ... OK

Name of the remote user of TARGET ISPConfig3 system: migrate
Password for this remote user: xxxxxxx
URL to the remote system (e. g. https://yourdomain.com:8080/remote/):
https://192.168.145.96:8080/remote/
You gave me remote user 'migrate' with password 'xxxxxxx'
And I shall connect to 'https://192.168.145.96:8080/remote/'
Is this correct? (y, n):

I will now have to make some changes in your ISPConfig3 TARGET system
config.
        1.) Enable domain module if needed
```

```
        2.) Remove client prefixes (will be undone automatically at the
end)
        3.) Enable login for mail accounts if needed

What is the main public(!) ip of the target server
webserver.dailydata.net? [127.0.1.1]: 192.168.145.96
What is the main public(!) ip of the target server
mailserver.dailydata.net? [127.0.1.1]: 192.168.145.97
What is the main public(!) ip of the target server
dnsserver.dailydata.net? [127.0.1.1]: 192.168.145.98

Seems you have a multi-server setup. So you have to choose target
servers for data now.
        [0] Don't import entries of this type
        [1] webserver.dailydata.net, main ip: 192.168.145.96
        [2] mailserver.dailydata.net, main ip: 192.168.145.97
        [3] dnsserver.dailydata.net, main ip: 192.168.145.98
Which server is the MASTER server of your multiserver setup? (0, 1, 2,
3): 1
[0] Don't import entries of this type
        [1] webserver.dailydata.net, main ip: 192.168.145.96
On which server you want the WEBSITES to be imported? (0, 1): 1
        [0] Don't import entries of this type
        [1] webserver.dailydata.net, main ip: 192.168.145.96
        [2] mailserver.dailydata.net, main ip: 192.168.145.97
        [3] dnsserver.dailydata.net, main ip: 192.168.145.98
On which server you want the DATABASES to be imported? (0, 1, 2, 3): 1
        [0] Don't import entries of this type
        [2] mailserver.dailydata.net, main ip: 192.168.145.97
On which server you want the MAIL ACCOUNTS to be imported? (0, 2): 0
        [0] Don't import entries of this type
        [3] dnsserver.dailydata.net, main ip: 192.168.145.98
On which server you want the DNS ENTRIES to be imported? (0, 3): 0
Do you want me to overwrite/update existing entries with the same name?
(y, n):
You can now set entry names, that shall NOT be overwritten.
You should use this if you plan to import several server's data to a
multi-server installation.
If this tool finds a name that is already present, it UPDATES the data
with the SOURCE data!
If this tool finds a name that you marked as not to overwrite, it
either
         - appends a _2, _3 etc. to this name and creates a new entry,
or
         - prepends a server prefix if specified by --server-prefix
argument
Do you want to change reserved names now? (y, n): n

WARNING! client templates can not be imported!
All imported clients will have their correct limits but no template
assigned!
```

```
Further things that will NOT be imported:
         - getmail accounts
         - APS information about installed APS packages

--- press ENTER if you have understood this and want to continue ---


Does your SOURCE ISPConfig 3 installation already use dovecot? [y]:

What is the database name of your SOURCE ISPConfig 3 installation?
[dbispconfig]:


Do you want to migrate only some services or everything?
Valid services are: client, web, mail, ftp, database, cronjob, dns,
billing
Please enter one or more services (comma-separated) or leave blank for
all: client,web,ftp,database,cronjob,billing
Should I try to keep the existing IDs for clients, webs etc. (IDs will
still change if old ID is taken already)? (y, n):
Calculating dependencies ...
All entries read.


To copy over web data, I need SSH access to the target webserver.
Please make sure that root login via SSH is allowed. On the target open
/etc/ssh/sshd_config
and set PermitRootLogin to yes or without-password.
You can revert this once the migration is complete.

What is the ip for the target web server to connect via SSH?
[192.168.145.96]:
What is the SSH port? [22]:

Testing MySQL connection ... OK
Testing target server's MySQL setting ...Warning
Warning Your max_allowed_packet setting is < 128M (16M). DB import
might fail.

Starting API calls.
[47/47] <Client> processing clientname (C20)
[67/67] <Domain> processing example.com
[67/67] <Website> processing example.co... www


[34/34] <DBUser> processing c6_example
[60/60] <Database> processing c6_example
```

```
Processing of entries done.


=============
Migration tool run completed.
```

Final Sync

At this point, all of your web and mail have been copied. However, the migration tool will **not** delete anything on the target machine; all mail deleted/moved over the migration period is on the Target mail server. In many cases, this is ok (your users will see mail they thought they had thrown away reappear), but a simple command will clean that up.

To ensure no e-mail is lost, first shut down all mail services on the Source mail server and disable restart:

```
service postfix stop
service dovecot stop
# For non-systemD systems
update-rc.d postfix disable
update-rc.d dovecot disable
# For systemD systems
# systemctl disable postfix
# systemctl disable dovecot
```

Now, from the Source machine, do the following after changing *Target* to the IP/Hostname of your target mail server. **Note** the first time does a *dry run*, ie shows you what will happen. After seeing it is working correctly, remove the *-dry-run* parameter.

```
rsync --dry-run --archive --verbose --delete --bwlimit=100m --exclude
'/mailfilters/' --exclude '/postqueue/' --exclude '/quota.group' --exclude
'/quota.user' --exclude 'sieve/' /var/vmail/* Target:/var/vmail/
```

This will show you what actions would be taken. When you are sure this is what you want, remove the –dry-run

Target becomes Source

Shut down the *Source* machines. For each of the *Target* machines, change the IP's as per https://www.howtoforge.com/community/threads/changing-ip-address-of-ispconfig-3-installation.52236/.

The following assumes you have the IP's set to 10.0.0.5-10.0.0.7 and you want to change them to 192.168.45.5-192.168.45.7.

On each server, run the following commands. This will find and change all references to the old IP's

and change them to the new IP's in any file in /etc. It also will save the original file to something with a .bak suffix.

Note that while the backslashes are not 100% needed in most cases, they do keep sed from accidentally changing something by mistake (the period is a wildcard, and the backslash escapes it so it does not match any character).

```
sed -ibak -e 's@\b10\.0\.0\.5\b@\b192\.168\.45\.5\b@g' /etc/*
sed -ibak -e 's@\b10\.0\.0\.6\b@\b192\.168\.45\.6\b@g' /etc/*
sed -ibak -e 's@\b10\.0\.0\.7\b@\b192\.168\.45\.7\b@g' /etc/*
cat /etc/network/interfaces # just make sure the IP is correct
```

Now, on the administrative server (the one you log into via the control panel), open the mysql database with:

```
mysql -u root -p dbispconfig
```

and run the following SQL commands:

```
UPDATE dns_rr SET DATA = '192.168.45.5' WHERE DATA = '10.0.0.5';
UPDATE dns_rr SET DATA = '192.168.45.6' WHERE DATA = '10.0.0.6';
UPDATE dns_rr SET DATA = '192.168.45.7' WHERE DATA = '10.0.0.7';
exit # TO leave the mariadb SQL server
```

If you are logged in from a console, you can now restart the network via:

We **may** have to set up mariDB permissions again so the servers can talk between themselves. I don't think so. I belive this is only for the initial installation. But, if your servers fail to communicate with each other (read syslog) do the following on the server with the web interface. Log into the database as the root user, then:

```
CREATE USER 'root'@'192.168.45.6' IDENTIFIED BY 'myrootpassword';
GRANT ALL PRIVILEGES ON * . * TO 'root'@'192.168.45.6' IDENTIFIED BY
'myrootpassword' WITH
GRANT  OPTION  MAX_QUERIES_PER_HOUR  0  MAX_CONNECTIONS_PER_HOUR  0
MAX_UPDATES_PER_HOUR  0
MAX_USER_CONNECTIONS 0 ;
CREATE USER 'root'@'192.168.45.7' IDENTIFIED BY 'myrootpassword';
GRANT ALL PRIVILEGES ON * . * TO 'root'@'192.168.45.7' IDENTIFIED BY
'myrootpassword' WITH
GRANT  OPTION  MAX_QUERIES_PER_HOUR  0  MAX_CONNECTIONS_PER_HOUR  0
MAX_UPDATES_PER_HOUR  0
MAX_USER_CONNECTIONS 0 ;
CREATE USER 'root'@'192.168.45.8' IDENTIFIED BY 'myrootpassword';
GRANT ALL PRIVILEGES ON * . * TO 'root'@'192.168.45.8' IDENTIFIED BY
'myrootpassword' WITH
GRANT  OPTION  MAX_QUERIES_PER_HOUR  0  MAX_CONNECTIONS_PER_HOUR  0
MAX_UPDATES_PER_HOUR  0
MAX_USER_CONNECTIONS 0 ;
```

```
service network restart
```

But, if you are not on a console, you will need to restart the server. **WARNING**: if you messed up on the commands, you will not be able to log back in on the new IP.

Recovery

I actually had the system abort on my first try, and it did not correctly reset the services on the *Source* server. Following were the steps I took to recover the system.

1. Import a copy of the dbispconfig database

   ```
   mysql -u root -p dbispconfig < /recent/backup/of/db
   ```

2. Open web interface to ISPConfig and resync all services
   1. **PUT REST OF STEPS HERE**
3. manually restart apache2 with command

   ```
   service apache2 restart
   ```

Links

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/software/controlpanels/ispconfig3/migration

Last update: **2019/08/05 00:25**