Using the oncocardio database with a different domain

Juan Domingo (Juan.Domingo@uv.es)

This guide is intended for system administrators who want to run the oncocardio database with the purpose of reproducibility or to get inspiration to build a similar system with a different database.

The simplest way to test the system is to use it by running the virtual machine we provide. It is a .vmdk disk image file that can be run with VMWare (either Player or Workstation) or with qemu/KVM. From now on, the virtual machine that is run in that way will be called the 'virtual server'.

Step 1 : Choose and register a machine name.

First, you will have to choose a virtual host name (in our case it was 'biodb') so the virtual server will be `yourname.your.domain` (in our case, `biodb.uv.es`). Ask your system administrators for an IP to be assigned to it. If you plan to configure the network of the virtual server by DHCP, generate randomly a MAC (but take note of it) and deliver it to your network administrators that will associate it to the new IP.

Step 2 : Prepare network in real host.

Network configuration has been thought to run the virtual server in a real machine with two or more network cards using one of them as a dedicated network interface for the virtual server. This is to prevent conflicts and to assign more easily a different IP number and domain to the virtual server.

The network interface intended to be used for the virtual server must be left untouched by the host (the real machine). To do so:

If you use NetworkManager : prevent NetworkManager from using such card. Directions to do so can be found for example in https://support.qacafe.com/knowledge-base/how-do-i-prevent-network-manager-from-controlling-an-interface/

If you do not use NetworkManager :

a) In Fedora/Rocky Linux/Redhat: do not add a `/etc/sysconfig/ifcfg-<interface>` for that interface.

b) In Debian/Ubuntu: Alter the `/etc/network/interfaces` according to the directions in https://wiki.debian.org/NetworkConfiguration

Step 3 : Create a virtual machine.

Now create a virtual machine with VMPlayer or qemu/KVM (with `virt-manager`). We recommend to assign it either one or two virtual CPUs, and about 2GB of RAM. The operating system obviously must not be installed since it is already installed, together will all the needed software, in the .vmdk image we provide (if VMWare asks for the version, it is a Fedora 32). This virtual disk is to be set as primary (and probably only) disk of the first virtual SATA interface.

The only detail to be specially changed in the creation of the virtual machines is the network configuration:

For creation with virt-manager : set as network source a "Macvtap device" and as device name, the name the host gives to the dedicated network card (in our case, it was `enp3s0` but find it with command `ip link show` or `ifconfig -a`). If you plan to boot the virtual server with dhcp, edit the XML and alter the MAC of this virtual interface to match the one your DHCP server expects. More information on this can be found in https://linuxconfig.org/how-to-use-bridged-networking-with-libvirt-and-kvm

For creation using VMWare : mark the network adapter of the virtual machine as 'Connect at power on' and 'Bridged' and, in the 'Advanced' part of the network configuration, set the MAC to the one expected by your DHCP server if you plan to use dhcp.

Step 4 : Virtual machine boot configuration (change of name and domain).

Now, boot the virtual server. It boots in multi-user mode (runlevel 3, no graphical interface). Log in as `root`, with password `biodbpwd`. Change the root password if you wish with command `passwd`.

From now on, all commands must be executed as the superuser (root).

Step 4a : Virtual DHCP network configuration.

The virtual server is currently prepared to boot and start the network using DHCP, so if you have used that and asked for a new IP, the IP should have been assigned as long as the machine name.

Step 4b : Alternative manual network configuration

If you do not use DHCP and prefer to configure the virtual host network manually, execute these commands:

```
cd /etc/sysconfig/network-scripts/
cp ./tmp/ifcfg-ens1 ./
nano ifcfg-ens1
```

This will open an editor to change the network configuration file. Only the values for `HWADDR, IPADDR, NETMASK` and `GATEWAY` should be altered with the correct values that you will have to ask to your network administrator (except the HWADDR, that must be the one you chose as MAC when creating the virtual machine). Finally, edit the file `/etc/hostname` and write there your machine name and domain.

After this, execute

```
systemctl stop NetworkManager
systemctl disable NetworkManager
systemctl restart network
```

Step 5 : Network check.

Whatever the way you have configured the network, you can check that it is working with commands

`systemctl status network` (which should inform of 'running' status)
`systemctl status NetworkManager` (as the previous)
`ip addr show` (which should show the MAC and IP address assigned to the network interface)
`hostname` (which should show the name of the machine)
`hostname -d` (which should show the domain of the machine)
`hostname -i` (which should show the IP address that must coincide with the one shown by the ip command before)

Step 6 : Acquisition of certificates.

To be able to operate in secure mode (encrypted hypertext transfer protocol, or https) you need to obtain the TLS certificates recognized by an external validation authority. Currently, Let's Encrypt (a nonprofit certificate authority ruled by the Internet Security Research Group) provides such certificates for free. Details on the procedure to set up can be found in https://certbot.eff.org/instructions.

Nevertheless, the provided virtual server has already installed the software to get and update such certificates. You only have to execute command

```
certbot certonly --nginx
```

and answer the questions about your server name, domain and organization. This will create the folder `/etc/letsencrypt/live/<your-server-name>` containing four files (in fact, symbolic links to files with the same name in `/etc/letsencrypt/archive/<your-server-name>`) named `cert.pem`, `chain.pem`, `fullchain.pem` and `privkey.pem`.

Step 7 : Setting up the server.

Now, edit the file `/etc/nginx/nginx.conf` and replace all references to `biodb.uv.es` by your virtual server name-domain. It appears five times, in lines which currently say

```
server_name biodb.uv.es;
ssl_certificate /etc/letsencrypt/live/biodb.uv.es/fullchain.pem; # managed by Certbot
ssl_certificate_key /etc/letsencrypt/live/biodb.uv.es/privkey.pem; # managed by Certbot
if ($host = biodb.uv.es)
server_name biodb.uv.es;
```

Step 8 : Generate internal certificates.

Use the newly obtained certificates to generate the certificates for the internal tomcat server. To do so, the commands are (substitute yourserver.your.domain by you particular value and biodbpwd by any password you choose. We suggest to use the same as for the root password of the virtual server):

```
cd /etc/certs/yourserver.your.domain
openssl pkcs12 -export -in fullchain.pem -inkey privkey.pem -out server.p12 \
-name tomcat
```

```
keytool -importkeystore -deststorepass biodbpwd -destkeypass biodbpwd \
-destkeystore /etc/tomcat/fkeystore.jks -srckeystore server.p12 \
-srcstoretype PKCS12 -srcstorepass biodbpwd -alias tomcat
rm server.p12
```

where the symbol \at the end of a line means 'follows in the same line'.

Step 9 : Update references to new server name.

Some configuration files must be changed to refer to you new host/domain name. In particular:

Substitute the references to `biodb.uv.es` by your server and domain in the following files:

`/home/dist/ocserv.js` (several references)
`/etc/tomcat/server.xml` (just one reference in line `keystoreFile="/etc/tomcat/biodb.jks"`)

and change appropriately according to your preferences the file

`/etc/oncocardio/mailtext`

which is the text sent by mail to the user when they receive their results attached.
Also, change and the file

`background.png`

which is the background the users see in the oncocardio screen.

Step 10 : Reboot

Just reboot the virtual machine with command `reboot`. Then you should be able to use a browser to access by https to your domain. The oncocardio database should work in your domain.