**Step by Step guide for deployment of new XNAT**

**Login Snapshots and Docker/Script locations:**

**Storage Server**

IP (Scan): 192.168.208.249

Username: xnatuser

Password: wakeupuser

**XNAT host**

Currently XNAT is on Samia's Local Machine.

To be moved to Pea3 or any of the other pea machines.

IP (Pea3): 192.168.208.132

Username: {your user}

Password: {your password}

**XNAT web browser login**

Login: admin

Password: admin

Please use an updated browser (Firefox or Chrome).

**XNAT Docker and Scripts**

XNAT docker location: /home/samia/Documents/xnat-docker-compose

XNAT Scripts: /mnt/wwn-0x5000c500cc87eb78/XNAT\_MIGRATION\_1.8.3

Original XNATPy: /mnt/wwn-0x5000c500cc87eb78/xnatpy

**How Restart Samia’s Machine for XNAT to work (sudo password: samia1234):**

Kill DBeaver

Check if postgres is running:

netstat -anp tcp | grep 5432

lsof -i tcp:5432

**# Stop a PostgreSQL service**

$ sudo service postgresql stop

# Stop xnat manually if you need to:

$ cd /home/samia/Documents/xnat-docker-compose

$ docker-compose down

**# If u want to delete the instance (this does not delete the data):**

$ docker-compose down –rmi all

If you want to delete the instance and delete all data from xnat:

$ docker-compose down –rmi all

$ sudo service postgresql stop

$ ssh [xnatuser@192.168.208.249,](mailto:xnatuser@192.168.208.249) Password: wakeupuser **(this gives you sudo access, BE CAREFUL!!)**

$ cd /mnt/data/XNAT-DATA

$ sudo rm –rf \*-data (this removes all data and postgres)

**# Unlink all mounts**

$ sudo umount –l /home/samia/Documents/xnatuser

$ sudo umount –l /home/samia/Documents/dgx1

$ sudo umount –l /home/samia/Documents/dgx2

$ sudo umount –l /home/samia/Documents/dgx3

$ sudo umount –l /home/samia/Documents/dgx4

$ sudo umount –l /home/samia/Documents/pea1

$ sudo umount –l /home/samia/Documents/pea2

$ sudo umount –l /home/samia/Documents/pea3

**# Shutdown and restart the computer**

$ sudo shutdown

**# Once the computer is up and running, check that the below folder is empty:**

$ ls /home/samia/Documents/xnatuser

**# Remount /scan**

$ sshfs -o allow\_other xnatuser@192.168.208.249:/mnt/data /home/samia/Documents/xnatuser

**# Go inside XNAT-migration-scripts folder:**

$ cd /mnt/wwn-0x5000c500cc87eb78/XNAT\_MIGRATION\_1\_8\_3

$ cd core

$ conda activate linkmood

Note: Only do run the below if you have deleted 192.168.208.249:/mnt/data/XNAT-DATA/xnat-data 192.168.208.249:/mnt/data/XNAT-DATA/postgres-data folders

$ python3.7 create\_xnat\_data\_folders.py (this creates the xnat-data and postgres-data inside /home/samia/Documents/xnatuser/XNAT-DATA (the new scan server)

**# Bring back xnat up**

$ cd /home/samia/Documents/xnat-docker-compose

$ docker-compose up –d

**# Once XNAT instance is running, you can check with:**

$ docker ps –l (this is small L)

**# Open Chrome or Firefox and type:**

URL: <https://localhost>

**# Login to XNAT:**

Username: admin

Password: admin

**# Create a project (New --> Project --> UCLH), if it is a clean instance or otherwise you should be able to check the existing data if you never deleted the data folders inside XNAT-DATA.**

**For migrating from old to new xnat:**

1. Run E:\SOFTWARE\Scripts\extract\_data\access\_xnat\_v3.py

python access\_xnat\_v3.py -incsv Unique\_XNAT\_identifiers\all\_unique\_ids.csv -o E:\DATA2\for\_xnat\_upload

This script has a line:

df = pd.read\_csv(args.incsv, low\_memory=False, nrows=50) #, skiprows=range(1, 100))

Please change the nrows manually to download paths to subject data in csvs that are stored at

E:\DATA2\for\_xnat\_upload

Add skiprows to not redownload the same paths when you run this script the next time.

Each downloaded csv file inE:\DATA2\for\_xnat\_upload has a timestamp, you need this to pass to /mnt/wwn-x5000c500cc87eb78/XNAT\_MIGRATION\_1\_8\_3/import\_data/upload.py --datetimestamp {value}

1. Mount the old server

sudo mount -t cifs //192.168.208.12/E /mnt/wwn-0x5000c500cc87eb78/xnat\_test -o username=Administrator

1. Run /mnt/wwn-x5000c500cc87eb78/XNAT\_MIGRATION\_1\_8\_3/import\_data/upload.py to start uploading new data to xnat.
2. You can check patients uploaded data from the browser http://localhost.

**Links to XNATpy, PyXNAT, DAX:**

<https://xnat.readthedocs.io/en/latest/static/tutorial.html#importing-data-into-xnat>

<https://github.com/pyxnat>

https://dax.readthedocs.io/en/latest/

**Important logins and commands stored at:**

/home/samia/Desktop/docker\_commands.txt

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1. **BRC2: Overview**

XNAT version: 1.6.5 (full installation)

Installed Components: Tomcat 7, PostgreSQL 9.4, XNAT WebAPP (NGINX)

Host machine: Windows 10 Release 2013

Python version: Python 2.7

Editor: Notepad ++

Storage Disk: E:\\

Physical data storage for XNAT: E:\\XNAT\_DATA\\archive

1.1 WebAPP:

1. Site URL: https://localhost:8080/brcii
2. Login: User, Password pre-saved on host machines Firefox browser.
3. Projects:

* DATA – contains all UCLH data be it dicom or niftis
* TEST – was created for testing RestAPI calls to XNAT
* BIOBANK – contains biobank niftis

(NOTE: biobank metadata is under schema biobank separately, see below)

1.2 PostgreSQL: Default installation, pgAdminIII installed

1. Hostname: localhost
2. Port: 5432
3. Database name: brcii
4. Xnat schema name: public
5. Number of tables: 697 (table creation automatic when xnat is installed)
6. Owners: brcii, brcii\_aux
7. Ownership privileges:

brcii – superuser

brcii\_aux – SELECT

Biobank (METADATA – that comes in a csv file):

1. Database name: brcii
2. Biobank metadata schema name: biobank
3. Tables:

|  |  |
| --- | --- |
| Table name | Number of rows |
| catagories | 1691 |
| data | 51143030 |

* 1. Quality Control

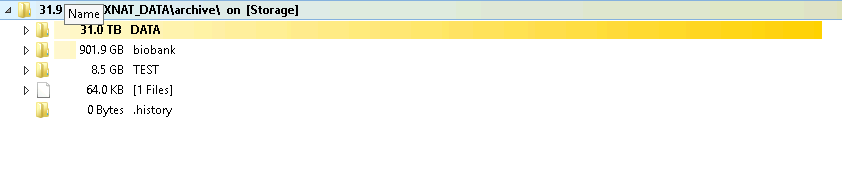
1. No specific quality control protocol followed. Everything flows directly to the archive via REST PUT calls.
2. Non-anonymised data objects. All information pertaining to the patients held in PostgreSQL including names, date of births and medical record numbers/hospital numbers.

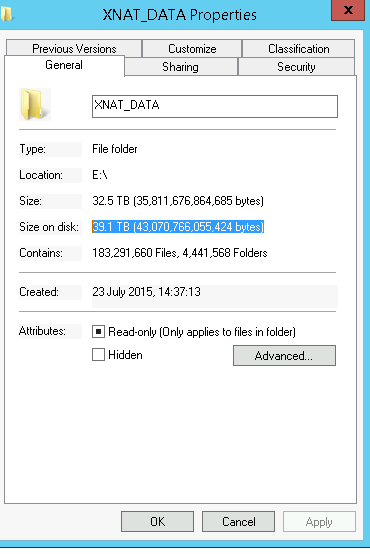
**Snapshot of Data in BRC2:**

Old brc server (Total space 181TB, 7TB free):

* Biobank Flair - 815.1 GB (to be replaced with latest downloads)
* Biobank T1 - 2TB (to be replaced with latest downloads)
* Neuro-Oncology GGM Data - 1.65TB (to be uploaded to new XNAT)
* Loose datasets (Tianbo's downloads) - To be updated with storage space info

Existing XNAT:





Migration plan of above:

- Move only DATA now to new XNAT because UCLH scans in existing XNAT are stored inside DATA.

- With the space left, move the loose datasets lying around in brc2.

- Lastly, make a Biobank project and strategize what to store.

New Server (Total space 139TB, 36TB free):

* Biobank Flair - 1.4TB (move to old brc)
* Biobank T1 - 2TB (move to old brc)
* Biobank DWI - 33.8TB
* Biobank Exome - 30TB (delete to make space/copy to tape)
* Biobank fMRI - 27.8TB (delete to make space/copy to tape)

**XNAT Upgrade:**

Computer Host machine:

* Architecture: Supermicro
* OS: Ubuntu 18.04
* CPUs: 20
* GPUs: NVIDIA VOLTA 100 x 4
* GPU memory: 32GB
* HDD: RAID, 7TB
* SSD: NVME, 500GB
* RAM: 256GB
* Connection IP: (Add here)
* Python version: 3x
* Docker support: Yes
* Docker-compose support: No (Need installation)
* GUI/Browser support: No (Need installation)

Storage Host machine:

* OS: Centos
* Storage Type: NVMESH, ZFS enabled low latency
* Storage Capacity: 139TB
* Connection IP: 192.168.208.249 (internal)
* Mounted on: /mnt/data

XNAT specifications:

* Version: 1.8.2.2 (Hotfix release)
* Installation: Dockerised XNAT
* Host Interfacing: Supermicro hosts the XNAT (IP), Physically XNAT data will be SCAN storage server (IP)
* Components: Tomcat 9, PostgreSQL 12.2, NGINX 1.19

WebAPP:

* 1. Site URL: <https://localhost>
  2. Port: 8080
  3. Login: Username - Admin, Password – Admin (GUI not enabled yet)
  4. Projects:
     + UCLH – contains all UCLH data be it dicom or niftis
     + BIOBANK – contains biobank niftis

**Data Migration Strategy:**

Best practice (Optional): Fill up a Change Request Window to list the changes to be made and enable a time window where the Proxy connection will be on for the host machines involved.

**Deploy XNAT on Supermicro:**

1. Install docker-compose into the host machine

2. Install Git if not installed.

3. Clone XNAT repository

$git clone <https://github.com/NrgXnat/xnat-docker-compose.git>

$cd xnat-docker-compose

$ cp default.env .env

In the docker-compose.yml (Major changes to Docker-compose.yml):

1. Create local directory to mount your sshfs drive

$ cd Documents

$ mkdir nfs\_mount

2. Fuse mount your sshfs drive

$ gedit /etc/fuse.conf

Uncomment user\_allow\_other

Save fuse.conf

$ sshfs -o allow\_otherxnatuser@192.168.208.249:/mnt/data/home/samia/Documents/xnatuser

3. Create all folders using file explorer (nautilus) inside the mounted drive:

* XNAT-DATA
  + xnat-data
    - archive
    - build
    - cache
    - home
      * logs
  + postgres-data

2. Give full permissions to these created folders with

* $ cd XNAT-DATA
* $ chmod 777 –R \*

Xnat-web: Change mount volumes

Volumes:

* /path to the disk where xnat-data will be created/xnat-data/home/logs: /data/xnat/home/logs
* /path to the disk where xnat-data will be created/xnat-data/archive:/data/xnat/archive

Xnat-db

Volumes:

* /path to the disk where postgres-data will be created/postgres-data: /var/lib/postgresql/data

NOTE: /path to the disk where xnat-data will be created/xnat-data/archive,

/path to the disk where postgres-data will be created/postgres-data MUST have “Read-Write” access (preferably chmod 777).

An example of working docker-compose.yml :

version: '3.7'

services:

xnat-web:

build:

context: ./xnat

args:

XNAT\_VERSION: ${XNAT\_VERSION}

XNAT\_SMTP\_ENABLED: ${XNAT\_SMTP\_ENABLED}

XNAT\_SMTP\_HOSTNAME: ${XNAT\_SMTP\_HOSTNAME}

XNAT\_SMTP\_PORT: ${XNAT\_SMTP\_PORT}

XNAT\_SMTP\_AUTH: ${XNAT\_SMTP\_AUTH}

XNAT\_SMTP\_USERNAME: ${XNAT\_SMTP\_USERNAME}

XNAT\_SMTP\_PASSWORD: ${XNAT\_SMTP\_PASSWORD}

XNAT\_DATASOURCE\_DRIVER: ${XNAT\_DATASOURCE\_DRIVER}

XNAT\_DATASOURCE\_URL: ${XNAT\_DATASOURCE\_URL}

XNAT\_DATASOURCE\_USERNAME: ${XNAT\_DATASOURCE\_USERNAME}

XNAT\_DATASOURCE\_PASSWORD: ${XNAT\_DATASOURCE\_PASSWORD}

TOMCAT\_XNAT\_FOLDER: ${TOMCAT\_XNAT\_FOLDER}

XNAT\_ROOT: ${XNAT\_ROOT}

XNAT\_HOME: ${XNAT\_HOME}

XNAT\_EMAIL: ${XNAT\_EMAIL}

ports:

- 8104:8104

volumes:

- ./xnat/plugins:${XNAT\_HOME}/plugins

- /home/samia/Documents/xnatuser/XNAT-DATA/xnat-data/home/logs:${XNAT\_HOME}/logs

- /home/samia/Documents/xnatuser/XNAT-DATA/xnat-data/archive:${XNAT\_ROOT}/archive

- /home/samia/Documents/xnatuser/XNAT-DATA/xnat-data/build:${XNAT\_ROOT}/build

- /home/samia/Documents/xnatuser/XNAT-DATA/xnat-data/cache:${XNAT\_ROOT}/cache

- /var/run/docker.sock:/var/run/docker.sock

depends\_on:

- xnat-db

environment:

- CATALINA\_OPTS=-Xms${XNAT\_MIN\_HEAP} -Xmx${XNAT\_MAX\_HEAP} -Dxnat.home=${XNAT\_HOME}

- XNAT\_HOME=${XNAT\_HOME}

xnat-db:

build:

context: ./postgres

args:

XNAT\_DATASOURCE\_PASSWORD: ${XNAT\_DATASOURCE\_PASSWORD}

ports:

- "5432:5432"

expose:

- "5432"

volumes:

- type: volume

source: postgresdata

target: /var/lib/postgresql/data

volume:

nocopy: true

environment:

POSTGRES\_PASSWORD: ${XNAT\_DATASOURCE\_ADMIN\_PASSWORD}

xnat-nginx:

build: ./nginx

ports:

- "80:80"

expose:

- "80"

links:

- xnat-web

volumes:

postgresdata:

driver\_opts:

type: "nfs"

o: "addr=192.168.208.249,nolock,soft,rw"

device: ":/mnt/data/XNAT-DATA/postgres-data"

4. XNAT environment variables can be changed in. env

5. Start the system:

$ docker-compose up –d

6. Open <http://localhost> in a browser and check if the web-app loads. It takes >= 1 minute for the web-app to load.

**XNAT project creation:**

Plan – Two projects to be created

a) Biobank – Will store all Biobank data that has been downloaded till date.

b) UCLH – Will store all hospital grade data in the form of NIFTIs or DICOMS as required/stored in current brc2 server.

Steps for Project Creation:

1. Go to XNAT web-app at <http://localhost>. First login will automatically redirect to Admin settings page. Make necessary changes and save settings.

2. At the XNAT home page, select Projects and then create the above projects.

Access permissions –

a) Biobank – Private

b) UCLH – Private

**Functional Testing: Test the interfacing of XNAT on supermicro and Physical Data Storage Server.**

Data upload:

1. From the Home page in XNAT web-app, select Upload. Select a Zipped Data file, select a project and click upload.

2. Deploy the XnatPy Docker. Check connection with XNAT localhost. Follow the instructions in the script to upload a data file.

Data Download:

1. Use the download script to download the single data file uploaded from XNAT to the local machine.

Data Delete:

1. Use Delete options inside XNAT to delete any files. This enables cascade deletes from PostgreSQL for the metadata.

**Static Tables Migration:**

1. Create all tables that store the Oncology data and the Stroke data inside the public schema.

2. Insert the downloaded (csv) via a python script.

This data is only present inside PostgreSQL.

**Migration UCLH:**

1. Mount E:\\ drive from 192.168.208.12 to the supermicro.

2. Access Pre-downloaded stored in E:\\DATA2\\for\_xnat\_upload.

3. Use upload script and pick the paths of the NIFTIS/DICOMS from the csv and push it to the new XNAT. The script will allow metadata to move to PostgreSQL.

4.Subject specific other information for the below tables need to be pushed by explicitly connecting to the PostgreSQL.

* public.xnat\_projectdata
* public.xnat\_subjectdata
* public.xnat\_experimentdata
* public.xnat\_imagesessiondata
* public.xnat\_imagescandata
* public.xnat\_mrscandata
* public.xnat\_subjectassessordata
* public.xnat\_abstractdemographicdata
* public.xnat\_abstractresource
* public.xnat\_demographicdata
* public.xnat\_resourcecatalog
* public.xnat\_resource

**Anonymization protocol:**

1. Create a table to hold hashes and original names, hospital numbers (mrns), date of birth of all subjects.

CREATE TABLE IF NOT EXISTS xnat\_hashoriginalmap( name varchar(255), patientId varchar(255), dateofbirth varchar(255), patientidentifier text, dateofbirthidentifier text);

1. Insert into these table the information selected from public.xnat\_imagesessiondata. All patientids are capitalized.

insert into xnat\_hashoriginalmap (patientId, name, dateofbirth) select upper(xi.dcmpatientid), xi.dcmpatientname, xi.dcmpatientbirthdate from xnat\_imagesessiondata xi where xi.dcmpatientid is not null and xi.dcmpatientname is not null and xi.dcmpatientbirthdate is not null group by xi.dcmpatientid, xi.dcmpatientname, xi.dcmpatientbirthdate;

3. Update patientId column in xnat\_imagesessiondata and xnat\_hashoriginalmap to remove all spaces.

update xnat\_hashoriginalmap set patientId = regexp\_replace(patientId, '\s+', '','g');

update xnat\_imagesessiondata set dcmpatientid = regexp\_replace(dcmpatientid, '\s+', '','g');

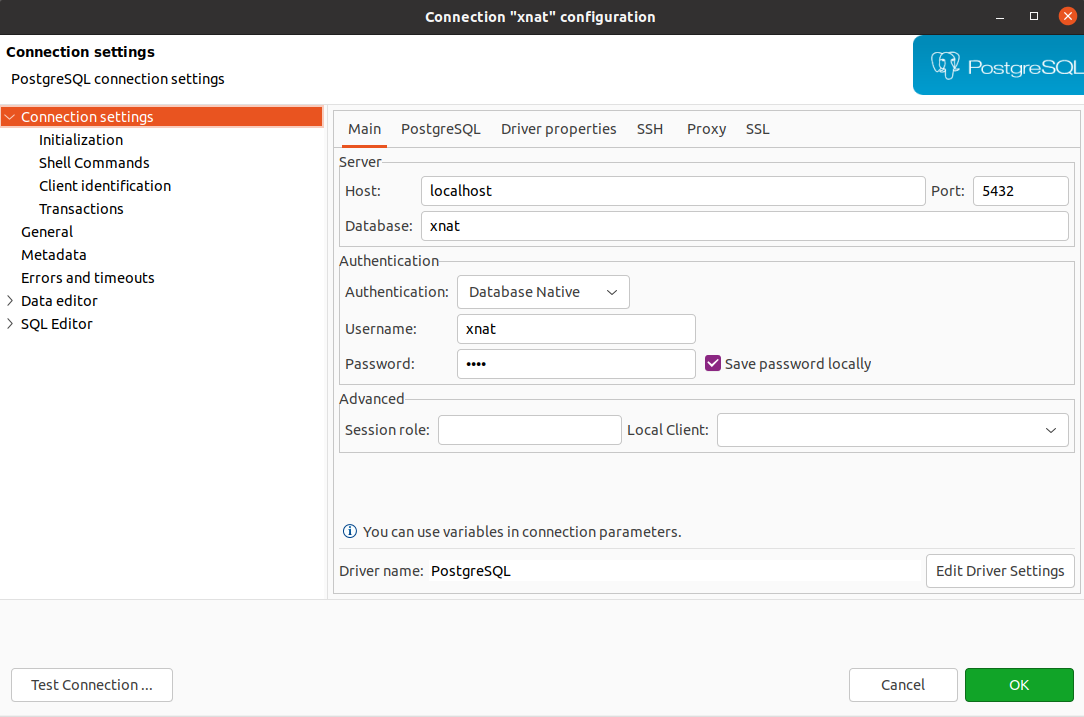
4. Read data from this table via anonymise hash python script. This script should produce a hash of 30 characters for the MRN and the dob.

5. Create hashdcmpatientbirthdate column in xnat\_imagesessiondata to store the hashes

alter table public.xnat\_imagesessiondata add hashdcmpatientbirthdate varchar(255);

6. update the hashes in xnat\_imagesessiondata table using anonymise\_hash.py script.

**Login to PostgresSQL from Dbeaver or psycopg2**



```python

import psycopg2

conn = psycopg2.connect(

host="0.0.0.0",

port="5432",

database="xnat",

user="xnat",

password="xnat"

)

```

As of now, I might have to do a manual PostgreSQL data move by

$docker-compose exec xnat-db psql -U xnat -h localhost -p 5432

The tables pre-exist inside the public schema. We must connect to the "xnat" database and fire the DML (insert, update) statements to these tables while copying the downloaded metadata from the csvs inside E:\\DATA2\\for\_xnat\_upload.

6. Download data: If user provides a hospital number, lookup the hash inside the hash-original mapping table. If found, download the files and respective metadata (optional). If not found, display ERROR: Subject not in XNAT.

**Mapping for Processing Postgres Inserts:**

|  |  |
| --- | --- |
| **Columns in BRC2 PostgreSQL** | **Columns in Exported CSV** |
| public.xnat\_mrscandata.xnat\_imagescandata\_id | image\_scandata\_id |
| public.xnat\_imagescandata.type | scan\_type |
| public.xnat\_imagescandata.image\_session\_id | image\_session\_id |
| public.xnat\_imagescandata.id | scan\_id |
| xnat\_experimentdata.label | session\_label |
| public.xnat\_subjectassessordata.subject\_id | subject\_id |
| public.xnat\_subjectdata.label | subject\_label |
| public.xnat\_imagescandata.series\_description | series\_desc |

**Migration Biobank (We are not migrating biobank due to crunch):**

1. All downloaded NIFTIs are stored inside the scan server in separate folders per Sequence.

2. Create Biobank subjects and sessions based on Dan Beasley's E:\\SOFTWARE\\Scripts\\import\_scripts\\ import\_biobank\_data.py. No anonymization required.

3. Use a biobank\_upload\_script.py to create all subjects, sessions and scans inside the Biobank XNAT project. Upload all NIFTIs to the Biobank project via this script.

4. Add the json files and other files for some Biobank subjects as subjectassessor resources.

5. Login to PostgreSQL. Create the metadata tables inside the public schema for Biobank.

6. Use a DML script that can add all metadata (categories and actual data values) to these tables. This is a one-time operation for now, however, these tables can grow later as new data comes through.

**Technology Options we have for Data Migrations:**

1. Distributed Automation of XNAT utilities for XnatUpload and XnatDownload.
2. XNATPy : Python scripts that have RESTAPI call structures under the hood. We can modify these scripts to make them suitable for UCLH and Biobank data movement. Currently, using XnatPy to upload and download.

Some points to investigate:

A) how to parallelise these REST API calls asynchronously, such that multiple subjects can be uploaded to XNAT at the same time without breaking.

B) how to integrate the PostgreSQL calls with these NIFTI upload calls. Currently, the PostgreSQL db, which is containerised, is unable to interact with any python calls from outside the environment.

3. Use direct REST calls such as PUT and GET.

**Major ISSUES:**

1. The PostgreSQL docker cannot be connected to from the outside with either python or Dbeaver. Have you added ports to your db service in the docker-compose.yml file?

Try adding

ports: '''Exposes the service to a localhost port that allows external connections'''

* + "5432:5432"

expose: '''Exposes the service to other dockers that are built together'''

* + "5432"

If the docker does not build after adding ports, check what is already running in 5432 with the below commands:

$netstat -anp tcp | grep 5432

$lsof -i tcp:5432

Stop if a PostgreSQL is already running with:

$sudo service postgresql stop

Try building the docker again.

Make sure your mount folder for the destination server is empty. Otherwise the docker will create a local folder and start storing data there. Do this:

sudo rm -rf '/home/samia/Documents/xnatuser/XNAT-DATA'

Next stop and remove the docker instance

$docker-compose down –rmi all

Stop all dockers if required:

$ sudo docker stop $(sudo docker ps -a -q)

$ sudo docker rm $(sudo docker ps -a -q)

$ sudo docker rm $(sudo docker ps -l -q)

**Docker-compose:**

Starting xnat-docker-compose\_xnat-db\_1 ... error

ERROR: for xnat-docker-compose\_xnat-db\_1 Cannot start service xnat-db: error evaluating symlinks from mount source "/var/lib/docker/volumes/xnat-docker-compose\_postgresdata/\_data": lstat /var/lib/docker/volumes/xnat-docker-compose\_postgresdata/\_data: stale NFS file handle

ERROR: for xnat-db Cannot start service xnat-db: error evaluating symlinks from mount source "/var/lib/docker/volumes/xnat-docker-compose\_postgresdata/\_data": lstat /var/lib/docker/volumes/xnat-docker-compose\_postgresdata/\_data: stale NFS file handle

ERROR: Encountered errors while bringing up the project.

samia@samia-SR-BX03866:~/Documents/xnat-docker-compose$ sudo umount -l /var/lib/docker/volumes/xnat-docker-compose\_postgresdata/\_data

**HELPFUL LINK**: https://stackoverflow.com/questions/38249434/docker-postgres-failed-to-bind- tcp-0-0-0-05432-address-already-in-useLINK:

On-going UCLH data notes:

1. No need to transfer any data from SubjectAssessorData because the only transferable column is 'age' and it is empty for all records in current xnat. Also, subjectid and id (experiment id/session label) are sequential self-generated columns.
2. Demographic Data cannot be inserted into latest xnat via postgres inserts because the hierarchical Foreign Key (FK) dependencies.

Solution: Use XnatPy to copy demographics per subject while uploading dicoms/niftis

Unique subjects to transfer data from Old brc to new brc:

90963

Anonymize these dicom tags (<https://forum.image-systems.biz/forum/index.php?user-post-list/132471-nikolag/>) :

(0029,1009) VERS="4" VR="LO" VM="1" Keyword="StudyDate" Name="Study Date"

(0029,1019) VERS="4" VR="LO" VM="1" Keyword="SeriesDate" Name="Series Date"

(0031,0010) VERS="4" VR="LO" VM="1" Keyword="PatientID" Name="Patient ID"

(0033,1004) VERS="4" VR="CS" VM="1" Keyword="StudyDescription" Name="Study Description"

(0033,1013) VERS="4" VR="PN" VM="1" Keyword="PatientName" Name="Patient’s Name"

(0010,0010), Patient’s Name

(0010,0020), Patient ID

(0010,0030), Patient’s Birth Date

(0010,1040), Patient’s Address

**Questions for DAN:**

* 1. How and where do you mount postgres-db folder for persistent storage?
  2. How do I bypass the firewall ufw installation specifics from your set up code and use it without it?
  3. What is the purpose of import-data-folder?
  4. Why do you use tomcat 8.5? Is it not recommended to use tomcat 9 with 1.8.2 since tomcat 8 versions cause instability?
  5. Why can it not build when archive path pre-exists? How is it supposed to persist data then? Can we not rebuild on the same path, if something goes wrong?
  6. Can you please explain what is happening under Migrating XNAT docker? What do you mean postgres folder cannot be mounted?

Xnat important links:

<https://wiki.xnat.org/container-service/configuring-a-container-host-126156926.html>