Install docker:

https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04

Install fmriprep

https://fmriprep.readthedocs.io/en/stable/installation.html#

Install dcm2bids

https://github.com/cbedetti/Dcm2Bids

Use dcm2bids

Place the DCM files in a directory
Create document.json. To do so:

- document.json includes the description of the data, see https://github.com/cbedetti/Dcm2Bids .
  The field “criteria” must be completed with the information of our data, which can be found by using the dcm2bids_helper (see previous link) and obtaining the information from the json files created. Finally the file should look like this:

```
{
   "descriptions": [
      {
         "dataType": "anat",
         "modalityLabel": "T1w",
         "criteria": {
            "SeriesDescription": "*T1*"
         }
      },
      {
         "dataType": "func",
         "modalityLabel": "bold",
                        "customLabels": "task-wm",
         "criteria": {
                                "SeriesDescription": "cmrr_*",
                                "MultibandAccelerationFactor": 3
         }
      },
        {
         "dataType": "func",
         "modalityLabel": "sbref",
                        "customLabels": "task-wm",
         "criteria": {
                                "SeriesDescription": "cmrr_FUNCIONAL_[Bb]*"
         }
      },
        {
         "dataType": "fmap",
         "modalityLabel": "phasediff",
         "criteria": {
                                "SeriesDescription": "gre_field_mapping",
                                "ImageType": ["ORIGINAL", "PRIMARY", "P", "ND"],
                                "SidecarFilename": "*_e2_ph.*"
         }
      },
        {
         "dataType": "fmap",
         "modalityLabel": "magnitude1",
         "criteria": {
                                "SeriesDescription": "gre_field_mapping",
                                "ImageType": ["ORIGINAL", "PRIMARY", "M", "ND", "NORM"],
                                "EchoTime": 0.00492,
                                "SidecarFilename": "*e1.*"
         }
      },
        {
         "dataType": "fmap",
         "modalityLabel": "magnitude2",
         "criteria": {
                                "SeriesDescription": "gre_field_mapping",
                                "ImageType": ["ORIGINAL", "PRIMARY", "M", "ND", "NORM"],
                                "EchoTime": 0.00738,
                                "SidecarFilename": "*e2.*"
         }
      }
   ]
}
```

*Before using it, delete sbref volumes, leaving just one of them.

From the directory where you want to export the BIDS files, type on console:
dcm2bids -d <DCM directory> -p <participantID> -s <sessionID> -c <document.json directory>
For example:

```
dcm2bids -d /archive/albamrt/MRI/DCM/E29/3 -p E29 -s 03 -c
/archive/albamrt/MRI/document.json
```

Check if your BIDS structure is correct at:
http://bids-standard.github.io/bids-validator/
If it's not, check errors, correct them and repeat step until correct format.

copy BIDS subject folder from DICOMS folder into new BIDS directory (fmriprep gets when run in dicom folder)

in func folder:

copy	sub-C01_ses-01_task-wm_run-1_sbref.json sub-C01_ses-01_task-wm_run-1_sbref.nii.gz
and rename them in	sub-C01_ses-01_task-wm_run-2_sbref.json sub-C01_ses-01_task-wm_run-2_sbref.nii.gz

Create phasediff.json, and place it in the directory where the BIDS will be placed. The file should contain echotimes for magnitude1 and magnitude2:

```
{
        "EchoTime1": 0.00492,
        "EchoTime2": 0.00738
}
```

Create task-wm_bold.json, and place it in the directory where the BIDS will be placed. The file contains information about the BOLD sequence:

```
{
        "TaskName": "wm",
        "RepetitionTime": 0.745,
        "EchoTime": 0.03,
        "FlipAngle": 90,
        "SliceTiming": [
                0,
                0.3925,
                0.0575,
                0.4475,
                0.1125,
                0.5025,
                0.1675,
                0.56,
                0.225,
                0.615,
                0.28,
                0.67,
                0.335,
                0,
                0.3925,
                0.0575,
                0.4475,
                0.1125,
                0.5025,
                0.1675,
                0.56,
                0.225,
                0.615,
                0.28,
                0.67,
                0.335,
                0,
                0.3925,
                0.0575,
                0.4475,
                0.1125,
                0.5025,
                0.1675,
                0.56,
                0.225,
                0.615,
                0.28,
                0.67,
                0.335  ],
        "MultibandAccelerationFactor": 3,
        "ParallelReductionFactorInPlane": 2,
        "PhaseEncodingDirection": "j-",
        "InstitutionName": "HOSPITAL_CLINIC_BARCELONA",
        "DeviceSerialNumber": "167044"
}
```

Create dataset_description.json, and place it in the directory where the BIDS will be placed. The file must include “Name” and “BIDSVersion”. Example:

```
{
   "Name": "NMDAR WM",
   "BIDSVersion":  "1.0.1"
}
```

Change the sub-C01_ses-01_phasediff.json in the fmap subdirectory to indicate all functional images in the “IntendedFor” field:

```
{
        "Modality": "MR",
        ...
                "IntendedFor": [
                "ses-01/func/sub-C01_ses-01_task-wm_run-01_bold.nii.gz",
                "ses-01/func/sub-C01_ses-01_task-wm_run-02_bold.nii.gz",
                "ses-01/func/sub-C01_ses-01_task-wm_run-01_sbref.nii.gz",
                "ses-01/func/sub-C01_ses-01_task-wm_run-02_sbref.nii.gz"],
        ...
}
```

The structure should be something like:

```
        ANMDA_BIDS
                dataset_description.json
                task-wm_bold.json
```

```
phasediff.json
sub-C01
        ses-01
                anat
                        sub-C01_ses-01_T1w.json
                        sub-C01_ses-01_T1w.nii.gz
                func
                        sub-C01_ses-01_task-wm_run-1_sbref.json
                        sub-C01_ses-01_task-wm_run-1_sbref.nii.gz
                        sub-C01_ses-01_task-wm_run-2_sbref.json
                        sub-C01_ses-01_task-wm_run-2_sbref.nii.gz
                        sub-C01_ses-01_task-wm_run-1_bold.json
                        sub-C01_ses-01_task-wm_run-1_bold.nii.gz
                        sub-C01_ses-01_task-wm_run-2_bold.json
                        sub-C01_ses-01_task-wm_run-2_bold.nii.gz
                fmap
                        sub-C01_ses-01_phasediff.nii.gz
                        sub-C01_ses-01_phasediff.json
                        sub-C01_ses-01_magnitude1.nii.gz
                        sub-C01_ses-01_magnitude1.json
                        sub-C01_ses-01_magnitude2.nii.gz
                        sub-C01_ses-01_magnitude2.json
        ses-02
        ...
sub-C02
...
```

Use MRIQC

https://mriqc.readthedocs.io/en/stable/

Use fmriprep

First, type in console:

```
sudo usermod -a -G docker $USER
```

Download Free Surfer license from:
https://surfer.nmr.mgh.harvard.edu/registration.html

Type in console:

```
fmriprep-docker --fs-license <directory/license.txt> <BIDS_directory>
<new_preprocessed_data_directory>
```

For example:

```
fmriprep-docker --fs-license /home/csc/Desktop/Docker_fmriprep/license.txt
/home/csc/Desktop/Docker_fmriprep/example/BIDSdir/
/home/csc/Desktop/Docker_fmriprep/example/preproc_dir/
```

```
fmriprep-docker --fs-license /archive/albamrt/MRI/license.txt /archive/albamrt/MRI/BIDS
/archive/albamrt/MRI/preprocess participant --participant_label C14 C15 C16 C17 C18
C19 C20 C21 -w /archive/albamrt/MRI/work -t wm --ignore slicetiming --bold2t1w-dof 6
--no-submm-recon --fs-no-reconall --write-graph -v --output-spaces
MNI152NLin6Asym:res-2 --write-graph --n_cpus 24 --nthreads 24
```

Check number of threads and CPUS depending on the node!!!

```
fmriprep-docker --fs-license /archive/albamrt/MRI/license.txt /archive/albamrt/MRI/BIDS
/archive/albamrt/MRI/preprocess participant --participant_label S15 -w
/archive/albamrt/MRI/work -t wm --ignore slicetiming --bold2t1w-dof 6 --no-submm-recon
--fs-no-reconall --write-graph -v --output-spaces MNI152NLin6Asym:res-2 --write-graph
--n_cpus 6 --mem-mb 20000
```

Actual command running, modified in order to have permission in output files (added -u $UID):

```
docker run --rm -it -e DOCKER_VERSION_8395080871=18.09.2 -v
/archive/albamrt/MRI/license.txt:/opt/freesurfer/license.txt:ro -v
/archive/albamrt/MRI/BIDS:/data:ro -v /archive/albamrt/MRI/preprocess:/out -v
/archive/albamrt/MRI/work:/scratch  -u $UID poldracklab/fmriprep:1.4.0 /data /out
participant --participant_label S23 -t wm --ignore slicetiming --bold2t1w-dof 6
--no-submm-recon --fs-no-reconall --write-graph -v --output-spaces
MNI152NLin6Asym:res-2 --write-graph --n_cpus 5 --nthreads 2 --omp-nthreads 4
--mem-mb 16000 -w /scratch --low-mem
```

Reading and writing from/to storage:

```
docker run --rm -it -e DOCKER_VERSION_8395080871=18.09.2 -v
/storage/albamrt/NMDA/MRI/license.txt:/opt/freesurfer/license.txt:ro -v
/storage/albamrt/NMDA/MRI/BIDS:/data:ro -v /storage/albamrt/NMDA/MRI/preprocess:/out
-v /storage/albamrt/NMDA/MRI/work:/scratch  -u $UID poldracklab/fmriprep:1.4.0 /data /out
participant --participant_label E18 -t wm --ignore slicetiming --bold2t1w-dof 6
--no-submm-recon --fs-no-reconall --write-graph -v --output-spaces
MNI152NLin6Asym:res-2 --write-graph --n_cpus 15 --omp-nthreads 4 --nthreads 4
--mem-mb 20000 -w /scratch --low-mem
```

```
# fluor:

docker run --rm -it -e DOCKER_VERSION_8395080871=18.09.2 -v
/archive/albamrt/MRI/license.txt:/opt/freesurfer/license.txt:ro -v
/archive/albamrt/MRI/BIDS:/data:ro -v /archive/albamrt/MRI/preprocess:/out -v
/archive/albamrt/MRI/work:/scratch  -u $UID poldracklab/fmriprep:latest /data /out
participant --participant_label E21 -t wm --ignore slicetiming --bold2t1w-dof 6
--no-submm-recon --fs-no-reconall --write-graph -v --output-spaces
MNI152NLin6Asym:res-2 --mem-mb 28000 -w /scratch --low-mem
```

```
docker run --rm -it -e DOCKER_VERSION_8395080871=18.09.2 -v
/archive/albamrt/MRI/license.txt:/opt/freesurfer/license.txt:ro -v
/storage/albamrt/NMDA/MRI/BIDS:/data:ro -v /storage/albamrt/NMDA/MRI/preprocess:/out
-v /storage/albamrt/NMDA/MRI/work:/scratch  -u $UID poldracklab/fmriprep:latest /data
/out participant --participant_label S17 -t wm --ignore slicetiming --bold2t1w-dof 6
--no-submm-recon --fs-no-reconall --write-graph -v --output-spaces
MNI152NLin6Asym:res-2 --mem-mb 28000 -w /scratch --low-mem
```

nitrogen :

```
docker run --rm -it -e DOCKER_VERSION_8395080871=18.09.2 -v
/archive/albamrt/MRI/license.txt:/opt/freesurfer/license.txt:ro -v
/archive/albamrt/MRI/BIDS:/data:ro -v /storage/albamrt/NMDA/MRI/preprocess:/out -v
/storage/albamrt/NMDA/MRI/work:/scratch  -u $UID poldracklab/fmriprep:latest /data /out
participant --participant_label E30 -t wm --ignore slicetiming --bold2t1w-dof 6
--no-submm-recon --fs-no-reconall --write-graph -v --output-spaces
MNI152NLin6Asym:res-2 --mem-mb 20000 -w /scratch --low-mem
```

```
docker run --rm -it -e DOCKER_VERSION_8395080871=18.09.2 -v
/archive/albamrt/MRI/license.txt:/opt/freesurfer/license.txt:ro -v
/archive/albamrt/MRI/BIDS:/data:ro -v /storage/albamrt/NMDA/MRI/preprocess:/out -w
/storage/albamrt/NMDA/MRI/work -u $UID poldracklab/fmriprep:latest /data /out participant
--participant_label E30 -t wm --ignore slicetiming --bold2t1w-dof 6 --no-submm-recon
--fs-no-reconall --write-graph -v --output-spaces MNI152NLin6Asym:res-2 --mem-mb
20000 --low-mem
```

```
/usr/local/miniconda/bin/fmriprep /data /out participant --participant_label C11 -t wm
--ignore slicetiming --bold2t1w-dof 6 --no-submm-recon --fs-no-reconall --write-graph -v
--output-spaces MNI152NLin6Asym:res-2 --mem-mb 28000 -w /scratch --low-mem
```

```
docker run --rm -it -e DOCKER_VERSION_8395080871=18.09.2 -v
/home/albamrt/MRI/license.txt:/opt/freesurfer/license.txt:ro -v
/archive/albamrt/MRI/BIDS:/data:ro -v /home/albamrt/MRI/preprocess:/out -v
/home/albamrt/MRI/work:/scratch  -u $UID poldracklab/fmriprep:latest /data /out participant
--participant_label C16 -t wm --ignore slicetiming --bold2t1w-dof 6 --no-submm-recon
--fs-no-reconall --write-graph -v --output-spaces MNI152NLin6Asym:res-2 --mem-mb
20000 -w /scratch --low-mem
```

```
docker run --rm -it -e DOCKER_VERSION_8395080871=18.09.2 -v
/home/albamrt/MRI/license.txt:/opt/freesurfer/license.txt:ro -v
/archive/albamrt/MRI/BIDS:/data:ro -v /home/albamrt/MRI/preprocess:/out -v
/home/albamrt/MRI/work:/scratch  -u $UID poldracklab/fmriprep:latest /data /out participant
--participant_label C12 -t wm --ignore slicetiming --bold2t1w-dof 6 --no-submm-recon
--fs-no-reconall --write-graph -v --output-spaces MNI152NLin6Asym:res-2 --mem-mb
20000 -w /scratch --low-mem --nthreads 1
```

```
--mem_mb 8000
```

--use-plugin /archive/albamrt/MRI/plugin.yml

```
usage: fmriprep-docker bids_dir output_dir [-h] [--version] [--skip_bids_validation]
       [--participant_label C01] list of participant IDs
       [-t TASK_ID] name of task to be processed (wm) [--echo-idx ECHO_IDX]
       [--nthreads NTHREADS] how many CPUs to use in parallel processing [--omp-nthreads
       OMP_NTHREADS] [--mem_mb MEM_MB] [--low-mem] [--use-plugin USE_PLUGIN]
       [--anat-only] [--boilerplate] [--ignore-aroma-denoising-errors]
       [--ignore {fieldmaps,slicetiming,sbref} [slicetiming]] no STC
       [-v] verbose [--longitudinal] [--t2s-coreg]
       [--bold2t1w-dof {6,9,12} 6] let's try with 6 first, if crappy increase
       [--output-space {T1w,template,fsnative,fsaverage,fsaverage6,fsaverage5} template]
       normalize to template provided by -template argument [--force-bbr]
       [--force-no-bbr]
       [--template {MNI152NLin2009cAsym}] [--output-grid-reference OUTPUT_GRID_REFERENCE]
       [--template-resampling-grid native] reference grid for EPI resampling
       [--medial-surface-nan] [--use-aroma] [--aroma-melodic-dimensionality
       AROMA_MELODIC_DIMENSIONALITY] [--skull-strip-template {OASIS,NKI}]
       [--skull-strip-fixed-seed] for run-to-run replicability skullstripping
       [--fmap-bspline] [--fmap-no-demean] [--use-syn-sdc] [--force-syn]
       [--fs-license-file PATH]
       [--no-submm-recon] disable submillimeter reconstruction [--cifti-output]
       [--fs-no-reconall] disable Freesurfer surface preprocessing
       [-w WORK_DIR] where to store intermediate results
       [--resource-monitor] [--reports-only] [--run-uuid RUN_UUID]
       [--write-graph] write workflow graph [--notrack]
       [--stop-on-first-crash] for debugging this might a good idea for now
       [--sloppy] low-quality preprocessing for testing, increases speed
```

Using fmriprep from the cluster – Create a singularity docker:

http://www.thememolab.org/resources/2018/02/05/running-bidsapps-on-cluster/