In [1]:
```
import cloudknot as ck
```

In [2]:
```
def fs_segment(subject):
    import os
    import os.path as op
    import boto3
    import numpy as np
    import nibabel as nib
    import nipype
    from subprocess import call
    resource = boto3.resource('s3')
    bucket_name = 'arokem.mri2mri'
    b = resource.Bucket(bucket_name)
    ll = list(b.objects.all())
    client = boto3.client('s3')

    print("Downloading T1 and DTI data")
    for l in ll:
        k = l.key
        if k.startswith("IXI-T1") and k.split('/')[-1].startswith(subject):
            print("Downloading T1")
            b.download_file(k, "/home/cloudknot-user/T1.nii.gz")

    FREESURFER_HOME = "/opt/freesurfer-6.0.0/"
    SUBJECTS_DIR = "/home/cloudknot-user/subjects/"
    os.mkdir(SUBJECTS_DIR)

    os.environ['SUBJECTS_DIR'] = SUBJECTS_DIR
    os.environ['FREESURFER_HOME'] = FREESURFER_HOME
    os.environ['MNI_DIR'] = FREESURFER_HOME + "mni"
    os.environ["FSFAST_HOME"] = FREESURFER_HOME + "fsfast"
    os.environ["OLDPWD"]= FREESURFER_HOME
    os.environ["FSF_OUTPUT_FORMAT"] = "nii.gz"
    os.environ["LOCAL_DIR"] = FREESURFER_HOME + "local"
    os.environ["FMRI_ANALYSIS_DIR"] = FREESURFER_HOME + "fsfast"
    os.environ["PWD"] = "/root"
    os.environ["FUNCTIONALS_DIR"] = FREESURFER_HOME + "sessions"
    os.environ["HOME"]="/root"
    os.environ["MNI_PERL5LIB"] = FREESURFER_HOME + "mni/share/perl5"
    os.environ["MINC_BIN_DIR"] = FREESURFER_HOME + "mni/bin"
    os.environ["PERL5LIB"]= FREESURFER_HOME + "mni/share/perl5"
    os.environ["FS_OVERRIDE"] = "0"
    os.environ["PATH"]= "/opt/freesurfer-6.0.0/bin:/opt/freesurfer-6.0.0/fsfast/bin
    os.environ["MINC_LIB_DIR"]= FREESURFER_HOME + "mni/lib"
    os.environ["MNI_DATAPATH"]= FREESURFER_HOME + "mni/data"


    from nipype.interfaces.freesurfer import ReconAll
    reconall = ReconAll()
    reconall.inputs.subject_id = subject
    reconall.inputs.directive = 'all'
    reconall.inputs.subjects_dir = SUBJECTS_DIR
    reconall.inputs.T1_files = "/home/cloudknot-user/T1.nii.gz"
    reconall.run()
```

```
    def upload_directory(path, bucketname, folder):
        for root, dirs, files in os.walk(path):
            for file in files:
                print(op.join(folder, os.path.split(root)[-1], file))
                client.upload_file(op.join(root, file),
                                   bucketname,
                                   os.path.join(folder, os.path.split(root)[-1], fi

    upload_directory(SUBJECTS_DIR + "%s/"%subject,
                     'arokem.mri2mri',
                     'IXI-Freesurfer-segmentations/%s/'%subject)
```

In [3]:
```
image = ck.DockerImage(func=fs_segment,
                       base_image="arokem/fs:2")
```

In [4]:
```
from glob import glob
```

In [5]:
```
ids = list(set([foo.split('/')[-1].split('-')[0] for
                foo in glob('/Users/arokem/data/mri2mri/t1_pdd_cosine_L1_unet128_T3
```

In [7]:
```
ids = ["IXI398",
    "IXI286",
    "IXI056",
    "IXI490",
    "IXI094",
    "IXI498",
    "IXI368",
    "IXI128",
    "IXI397",
    "IXI216",
    "IXI285",
    "IXI265",
    "IXI488",
    "IXI136",
    "IXI567",
    "IXI381",
    "IXI161",
    "IXI160",
    "IXI308",
    "IXI537",
    "IXI623",
    "IXI127",
    "IXI558",
    "IXI297",
    "IXI455",
    "IXI598",
    "IXI236",
    "IXI484",
    "IXI335",
    "IXI389",
    "IXI211",
    "IXI377",
    "IXI169",
```

```
    "IXI480",
    "IXI048",
    "IXI440"]
```

In [8]:
```
len(ids)
```

Out[8]: 36

In [9]:
```
knots = []
results = []
for i in range(len(ids) // 10 + 1):
    knots.append(ck.Knot(name='fs_seg_2_32_%s'%i,
                docker_image=image,
                #memory=10000,
                bid_percentage=100,
                resource_type="SPOT",
                #image_id="ami-0b251dc6f872712a4",
                pars_policies=('AmazonS3FullAccess',)))
    results.append(knots[-1].map(ids[i*10:(i+1)*10]))
```

In [ ]:
```
```