

How to create an EC2 instance ?

Go to <https://aws.amazon.com/console/>

Use your AWS username and password to login

Select EC2

The screenshot shows the AWS Services console. At the top left, there's a search bar with placeholder text "Example: Relational Database Service, database, RDS". Below it, under "Recently visited services", are links for EC2, Billing, IAM, AWS Cost Explorer, and S3. A section titled "All services" contains a tree view with Compute expanded, showing EC2 selected. Other collapsed categories include Lightsail, Lambda, and Batch. To the right of the tree view are links for Blockchain, Satellite, Security, Identity, & Compliance, and other services like IAM and Resource Access Manager.

Select launch instance

The screenshot shows the "Launch instance" wizard. It displays various resources: Dedicated Hosts (0), Volumes (0), Key pairs (6), and Placement groups (0). Below this is a note: "Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. [Learn more](#)".

The screenshot shows the "Launch instance" wizard. It has two main sections: "Launch instance" and "Launch instance from template". The "Launch instance" section includes a link to "Amazon Machine Image (AMI) Library". Below these sections is a note: "An Amazon Elastic Compute Cloud (Amazon EC2) instance, which is a virtual server in the cloud."

Select Deep Learning AMI (Amazon Linux 2) Version 27.0 instance

1. Choose Instance Type 2. Choose an Amazon Machine Image (AMI) 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Choose an Amazon Machine Image (AMI)

- Deep Learning AMI (Ubuntu 16.04) Version 27.0 - ami-0a79b70001264b442**
MXNet-1.6.0, Tensorflow-2.1.0 & 1.15.2, PyTorch-1.4.0, Keras-2.2, & other frameworks, configured with Neuron, NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker & NVIDIA-Docker. For fully managed experience, check: <https://aws.amazon.com/sagemaker>
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
- Deep Learning AMI (Amazon Linux 2) Version 27.0 - ami-03d7bb62671766e1e**
Amazon Linux
MXNet-1.6.0, Tensorflow-2.1.0 & 1.15.2, PyTorch-1.4.0, Keras-2.2, & other frameworks, configured with Neuron, NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker & NVIDIA-Docker. For fully managed experience, check: <https://aws.amazon.com/sagemaker>
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
- Deep Learning Base AMI (Ubuntu 18.04) Version 22.0 - ami-0f6127e61a87f8677**
Comes with foundational platform of NVidia CUDA, cuDNN, NCCL, GPU Drivers, Intel MKL-DNN and other system libraries to deploy your own custom deep learning environment. For a fully managed experience, check: <https://aws.amazon.com/sagemaker>
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
- Deep Learning Base AMI (Ubuntu 16.04) Version 22.0 - ami-09ea06f2ec4506f69**
Comes with foundational platform of NVidia CUDA, cuDNN, NCCL, GPU Drivers, Intel MKL-DNN and other system libraries to deploy your own custom deep learning environment. For a fully managed experience, check: <https://aws.amazon.com/sagemaker>

Select General purpose free tier eligible instance and click Review and Launch

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

2: Choose an Instance Type

Previously selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

| Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support |
|-----------------|--|-------|--------------|-----------------------|-------------------------|---------------------|--------------|
| General purpose | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderate | Yes |
| General purpose | t2.micro Free tier eligible | 1 | 1 | EBS only | - | Low to Moderate | Yes |
| General purpose | t2.small | 1 | 2 | EBS only | - | Low to Moderate | Yes |
| General purpose | t2.medium | 2 | 4 | EBS only | - | Low to Moderate | Yes |
| General purpose | t2.large | 2 | 8 | EBS only | - | Low to Moderate | Yes |
| General purpose | t2.xlarge | 4 | 16 | EBS only | - | Moderate | Yes |
| General purpose | t2.2xlarge | 8 | 32 | EBS only | - | Moderate | Yes |
| General purpose | t3a.nano | 2 | 0.5 | EBS only | Yes | Up to 5 Gigabit | Yes |
| General purpose | t3a.micro | 2 | 1 | EBS only | Yes | Up to 5 Gigabit | Yes |

Cancel Previous **Review and Launch** Next: Configure Instance Details

In the next screen, choose Create a new key pair and give a name to the key "free-tier-instance-aws-key". Select Download Key Pair and save it in your local drive. Later you will need this key to connect to EC2 instance.

Select an existing key pair or create a new key pair

X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

free-tier-instance-aws-key

Download Key Pair

You have to download the **private key file (*.pem file)** before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel

Launch Instances

Click Launch instance

Copy the Public DNS (IPv4)

| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Public DNS (IPv4) | IPv4 Public IP |
|------|---------------------|---------------|-------------------|----------------|---------------|--------------|-------------------------|----------------|
| | i-0b2333f74517ce90a | t2.micro | us-east-1a | running | Initializing | None | ec2-54-172-68-170.co... | 54.172.68... |
| | i-0bc0c509c300e1966 | t2.micro | us-east-1a | terminated | | None | | - |

Instance: i-0b2333f74517ce90a Public DNS: ec2-54-172-68-170.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID: i-0b2333f74517ce90a
Instance state: running
Instance type: t2.micro
Finding: Opt-in to AWS Compute Optimizer for recommendations. [Learn more](#)
Private DNS: ip-172-31-86-162.ec2.internal
Private IPs: 172.31.86.162

Secondary private IPs
VPC ID: vpc-467a573c
Subnet ID: subnet-949a86ba

Public DNS (IPv4): ec2-54-172-68-170.compute-1.amazonaws.com
IPv4 Public IP: 54.172.68.170
IPv6 IPs: -
Elastic IPs: -
Availability zone: us-east-1a
Security groups: launch-wizard-50, view inbound rules, view outbound rules
Scheduled events: No scheduled events
AMI ID: Deep Learning AMI (Amazon Linux 2) Version 27.0 (ami-03d7bb62671766e1e)
Platform details: -

11) Set key permission

```
chmod 400 free-tier-instance-aws-key.pem
```

12) Connect to EC2 instance

```
scp -i free-tier-instance-aws-key.pem ec2-user@ec2-54-172-68-170.compute-1.amazonaws.com
```

13) Copy files from local drive to EC2 instance

```
scp -i free-tier-instance-aws-key.pem test.txt ec2-user@ec2-54-172-68-170.compute-1.amazonaws.com:
```

14) Copy folders from local drive to EC2 instance

```
scp -i free-tier-instance-aws-key.pem -r model_folder ec2-user@ec2-54-172-68-170.compute-1.amazonaws.com:
```

15) Click on the below link . The link has instructions for installing the brain masking pipeline in linux machines.

<https://github.com/pnlbwh/CNN-Diffusion-MRIBrain-Segmentation>

If you face issues during automated installation, please consider manual installation

- conda create -n cnn_gpu python=3.6
- source activate cnn_gpu
- conda install tensorflow-gpu==1.12.0
- conda install numpy==1.16.4
- conda install keras==2.2.4
- conda install -c pnlbwh ants
- pip install nibabel
- pip install gputil
- pip install git+<https://github.com/pnlbwh/conversion.git>
- export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/home/ec2-user/anaconda3/envs/cnn_gpu/lib
- export ANTSPATH=/home/ec2-user/anaconda3/envs/cnn_gpu/bin

Please edit your `.bashrc` file and add the following 3 lines.

- export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/home/ec2-user/anaconda3/envs/cnn_gpu/lib
- export ANTSPATH=/home/ec2-user/anaconda3/envs/cnn_gpu/bin
- source activate ~/anaconda3/envs/cnn_gpu

Running 4 brain masking scripts

```
python extractb0.py -i /home/ec2-user/aws_cases/cases.txt  
python antsRegistration.py -i /home/ec2-user/aws_cases/cases.txt -f /home/ec2-user/model_folder  
python maskprocessing.py -i /home/ec2-user/aws_cases/cases.txt -f /home/ec2-user/model_folder  
python postprocessing.py -i /home/ec2-user/aws_cases/cases.txt
```

Mount s3 bucket to aws ec2 instance:

- 1) Follow the below tutorial until step 8

<https://cloudkul.com/blog/mounting-s3-bucket-linux-ec2-instance/>

- 2) For step 9 use the following command on ec2 instance to mount s3 bucket

```
s3fs nda-enclave-c3371 /home/ec2-user/mys3bucket -o use_cache=/tmp -o  
allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 -o  
use_path_request_style -o url=https://s3-us-east-1.amazonaws.com
```

- 3) Run df -Th command in your ec2 terminal

```
(cnn_gpu) [ec2-user@ip-172-31-64-62 .aws]$ df -Th  
Filesystem      Type     Size   Used  Avail Use% Mounted on  
devtmpfs        devtmpfs  7.7G    0    7.7G  0% /dev  
tmpfs           tmpfs    7.7G    0    7.7G  0% /dev/shm  
tmpfs           tmpfs    7.7G  452K  7.7G  1% /run  
tmpfs           tmpfs    7.7G    0    7.7G  0% /sys/fs/cgroup  
/dev/nvme0n1p1  xfs     90G   75G   16G  83% /  
tmpfs           tmpfs    1.6G    0    1.6G  0% /run/user/1000  
s3fs            fuse.s3fs 90G   75G   16G  83% /home/ec2-user/mys3bucket
```

- 4) Check for any error message under less /var/log/messages

```
(cnn_gpu) [ec2-user@ip-172-31-64-62 ~]$ sudo s3fs nda-enclave-c3371 /home/ec2-user/mys3bucket -o use_cache=/tmp -o allow_other -o  
uid=1001 -o mp_umask=002 -o multireq_max=5 -o use_path_request_style -o url=https://s3-us-east-1.amazonaws.com  
(cnn_gpu) [ec2-user@ip-172-31-64-62 ~]$ df -Th  
Filesystem      Type     Size   Used  Avail Use% Mounted on  
devtmpfs        devtmpfs  7.7G    0    7.7G  0% /dev  
tmpfs           tmpfs    7.7G    0    7.7G  0% /dev/shm  
tmpfs           tmpfs    7.7G  452K  7.7G  1% /run  
tmpfs           tmpfs    7.7G    0    7.7G  0% /sys/fs/cgroup  
/dev/nvme0n1p1  xfs     90G   75G   16G  83% /  
tmpfs           tmpfs    1.6G    0    1.6G  0% /run/user/1000  
s3fs            fuse.s3fs 90G   75G   16G  83% /home/ec2-user/mys3bucket  
(cnn_gpu) [ec2-user@ip-172-31-64-62 ~]$ df -Th  
Filesystem      Type     Size   Used  Avail Use% Mounted on  
devtmpfs        devtmpfs  7.7G    0    7.7G  0% /dev  
tmpfs           tmpfs    7.7G    0    7.7G  0% /dev/shm  
tmpfs           tmpfs    7.7G  452K  7.7G  1% /run  
tmpfs           tmpfs    7.7G    0    7.7G  0% /sys/fs/cgroup  
/dev/nvme0n1p1  xfs     90G   75G   16G  83% /  
tmpfs           tmpfs    1.6G    0    1.6G  0% /run/user/1000  
(cnn_gpu) [ec2-user@ip-172-31-64-62 ~]$ sudo cat /var/log/messages | tail -5  
Mar 26 15:30:01 ip-172-31-64-62 systemd: Removed slice User Slice of root.  
Mar 26 15:30:01 ip-172-31-64-62 systemd: Stopping User Slice of root.  
Mar 26 15:31:19 ip-172-31-64-62 dhclient[3567]: XMT: Solicit on eth0, interval 121500ms.          error message  
Mar 26 15:32:40 ip-172-31-64-62 s3fs[4920]: init v1.86(commit:fe2b269) with OpenSSL  
Mar 26 15:32:51 ip-172-31-64-62 s3fs[4920]: s3fs.cpp:s3fs_check_service(3905): unable to connect(host=https://s3-us-east-1.amazonaws.com) - result of checking service.  
(cnn_gpu) [ec2-user@ip-172-31-64-62 ~]$
```

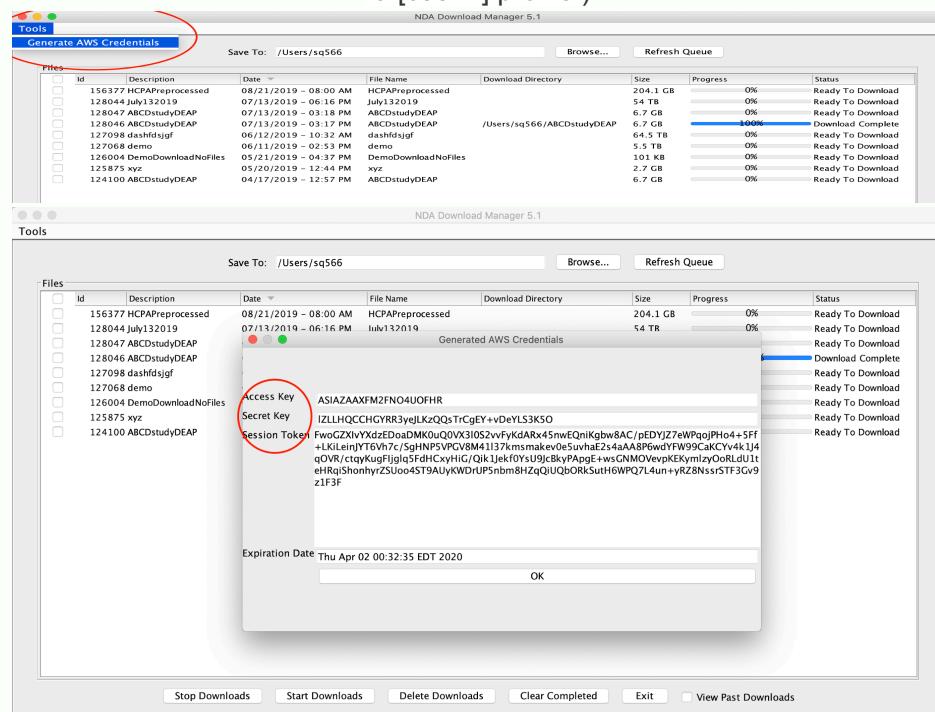
Manage aws credentials:

Instead of keeping credentials in environment variables, you can now put credentials into a single file that's in a central location. The default location is this: `~/.aws/credentials` (Linux/Mac)

```
(base) sq566@PHS010667:~/aws$ pwd
/Users/sq566/.aws
(base) sq566@PHS010667:~/aws$ ls
config      credentials
(base) sq566@PHS010667:~/aws$ cat credentials
[default]
aws_access_key_id = ASIAZAAAXFM2FB50TJUQW
aws_secret_access_key = 0y67alcwg20XghFhBsEta6zzIo/fHQZKMB0n0xe
aws_session_token = FwoGZXIVYXdzE0aDMZwBaLgoVYsms8U0iKdAfqrSmvzgZJ94LNkrSaT0fUa6tChi3eMIam6CXN9yFYQZBYP6TyWwif2Tm9qW+Erk7MfpFuzv5e
o2U0gA3rh4ZlwKBdPpEm1p3N27Z9AllD9LP1P81i4jZXNwSISbKKsHpuDQA0sfrMu+7Te/vE4w7IyKGiR20GEIDwmkm+zxtQwiQDBUYwCd4b2osaEPCBda73yCtFKY
RBevzQgorf+S9AUyKdD+QP8r0ibtk7nzXWYRqHFB+f01dTbvdUTCLqh10LSYQH8H2umb1

[c3371]
aws_access_key_id = ASIAZAAAXFM2FP5TFCN6X
aws_secret_access_key = lhSpb+YXM4KgM0NFz/HzeZc00DYTv10oFlJJsdh
aws_session_token = FwoGZXIVYXdzE0aDPLcuMkMVXlbFY50BSKdA0dtC7m/GcEk2lwfaL/QUX6xfabomL/yDOLINe7GjnbFH4qMkh0jm23QV8aFaGndR3hSqStFB
m5T20eyrFod9Qzl8aveSDVQ1nXZ7l6z2l/Zycnn0lhfuuinMKFZ/MMkrY5UHAfVjjaFnk5KTwIpvEdELkM+NMsrsQaJenhAkgZnL1IkbdNrR2YlQ/xX0guk3zesrPv0Im
uxBsr8ogICT9AUyKZY70ijnk5XY0+rUJKU4cnQGDeYmkS0azYXwLQkmwiLZTE8Mgk3M0Wt
(base) sq566@PHS010667:~/aws$ _
```

- You can get `key_id`, `access_key` and `session_token` from NDA Download Manager (**Note:** You should generate aws credentials 2 times, one for the [default] profile and another for [c3371] profile)



AWS S3 Commands

1) Copy folder from local to s3 bucket

```
aws s3 cp site01 s3://nda-enclave-c3371/ --recursive
```

2) Copy file from local to s3 bucket

```
aws s3 cp files.txt s3://nda-enclave-c3371/
```

3) Move file from s3 bucket to s3 bucket folder

```
aws s3 mv s3://nda-enclave-c3371/files.txt s3://nda-enclave-c3371/site01
```

4) Remove files in s3 bucket

```
aws s3 rm s3://nda-enclave-c3371/files.txt
```

5) Recursively copy new and updated files from the source directory to the destination

```
aws s3 sync s3://nda-enclave-c3371/home/ec2-user/mys3bucket
```

6) List files inside bucket

```
aws s3 ls s3://nda-enclave-c3371 --profile c3771
```

7) Sync local folder with s3 bucket

```
aws s3 sync test_abcd s3://nda-enclave-c3371
```

8) Sync s3 bucket folder with local folder

```
aws s3 sync s3://nda-enclave-c3371/test_abcd .
```

How did I run the brain masking pipeline in an ec2 gpu instance for 70 abcd subjects ?

- 1) Connect to the GPU EC2 instance
- 2) Install the brain masking pipeline conda modules
- 3) Set up the aws credentials
- 4) mkdir /home/ec2-user/test_abcd
- 5) cd /home/ec2-user/test_abcd
- 6) <https://github.com/SenthilCaesar/CNN-Sematic-Segmentation/blob/master/aws/copy.sh>
- 7) ./copy.sh
- 8) ls -l | *.tgz | awk '{print \$NF}' > tar_cases.txt
- 9) <https://github.com/SenthilCaesar/CNN-Sematic-Segmentation/blob/master/aws/unzip.sh>
- 10) ./unzip.sh
- 11) The next step is to create the dwi input case list
- 12) ll -d sub-NDARINV*/**/*.nii | awk '{print \$NF}' > abcd_tmp.txt
- 13) sed 's/^\/home\/ec2-user\/test_abcd/' abcd_tmp.txt > abcd.txt
- 14) **python** extractb0.py -i /home/ec2-user/test_abcd/abcd.txt
- 15) **python** antsRegistration.py -i /home/ec2-user/test_abcd/abcd.txt -f /home/ec2-user/model_folder/
- 16) **python** maskprocessing.py -i /home/ec2-user/test_abcd/abcd.txt -f /home/ec2-user/model_folder/
- 17) **python** postprocessing.py -i /home/ec2-user/test_abcd/abcd.txt
- 18) copy the results back to the s3 bucket.
- 19) I removed the original dwi volume because it takes a lot of space, I only need b0 and b0_masks for qc'ing
- 20) zip -r abcd_ec2.zip test_abcd/
- 21) **aws** s3 cp abcd_ec2.zip s3://nda-enclave-c3371/
- 22) Now connect to pnl-maxwell where you have slicesdir module
- 23) Again set up the aws credentials in pnl-maxwell machine
- 24) Copy the result from s3 bucket to pnl-maxwell machine
- 25) **aws** s3 cp s3://nda-enclave-c3371/abcd_ec2.zip .
- 26) **unzip** abcd_ec2.zip
- 27) Create b0 caselist and b0 masks for slicesdir qc'ing
- 28) ll -d sub-NDARINV*/**/_bse.nii.gz | awk '{print \$NF}' > b0.txt
- 29) ll -d sub-NDARINV*/**/_BrainMask.nii.gz | awk '{print \$NF}' > b0_masks.txt
- 30) <https://github.com/SenthilCaesar/CNN-Sematic-Segmentation/blob/master/aws/read.sh>
- 31) ./read.sh > qc.txt
- 32) **sed** 's/^/slicesdir -o /' qc.txt > slicesdir.sh
- 33) chmod 755 slicesdir.sh
- 34) ./slicesdir.sh

Python code to create site text files from dwiTable excel sheet

The code will create site1.txt, site2.txt, site3.txt.....site39.txt

```
import pandas as pd
from pandas import ExcelWriter
from pandas import ExcelFile

df = pd.read_excel('dwiTable.xlsx')

print("Column headings:")
print(df.columns)

abcd_sub = df['derived_files'].tolist()
abcd_site = df['new_site_id'].tolist()

d = dict()
for i in range(len(abcd_sub)):

    site_no = abcd_site[i]
    site_sub = abcd_sub[i]

    if site_no not in d:
        d[site_no] = []
    else:
        d[site_no].append(site_sub)

for key, value in list(d.items()):
    name = 'site_' + str(key) + '.txt'
    with open(name, 'w') as f:
        for item in value:
            f.write("%s\n" % item)
```

Other Useful scripts:

Dice Coefficient:

https://github.com/SenthilCaesar/CNN-Sematic-Segmentation/blob/master/data-preprocessing/dice_coeff.py

Convert nifti volume to *.png images

https://github.com/SenthilCaesar/CNN-Sematic-Segmentation/blob/master/data-preprocessing/Raunak_cv2.py