Project Setup

Benjamin Lee

Jennifer Crosby

Pauline McKim

30 November 2020

The purpose of group three's final project is to implement a machine learning model that classifies deep sea images. Images of deep-sea environment are classified as either interesting or not interesting based on the content of the photo. Implementation of this model was performed using AWS services as explained below.

Construction of AWS EC2 Instance

An AWS EC2 instance was constructed with the following settings:

- AMI: Deep Learning Base AMI (Ubuntu 18.04) Version 31.0 - ami-063f381b07ea97834, built with NVIDIA CUDA, cuDNN, NCCL, GPU Drivers, Intel MKL-DNN, Docker, NVIDIA-Docker and EFA support. For a fully managed experience, check: https://aws.amazon.com/sagemaker
- Instance type: g3.4xlarge
- Network Settings: Selected VPC build for project and public subnet option. If public subnet option is not selected, Cloud9 cannot connect to the EC2. Created IAM role that allows EC2 and S3 to connect (Amazon S3 FullAccess policy applied).
- Storage size: 100 GB
- Tags: Added a tag for Name = DeepSeaEC2
- Security Groups: Security group ports opened for the following:
    - SSH, TCP 22 (Default)
    - SSH, TCP 80 (Port 80 Entry)
    - Custom Traffic, TCP 8888 (for potential Jupyter Notebook implementation)
    - All Traffic, 0 – 65535 (Default SG)
    - HTTPS, TCP 443 (for potential Boto implementation)

Construction of S3 Bucket

An AWS S3 bucket was also created. The necessary files were manually uploaded to the S3 using the AWS website. The files uploaded include the relevant .PY files that train and implement the machine learning model, over a thousand deep sea images, and a CSV file containing the labels for the images.

Bridge Between EC2 and S3

The necessary .pem file was created. The private key of the .pem file was copy- pasted, and chmod 400 ran to secure it. A connection was made to the EC2 using SSH with the .pem key. AWS configure was applied with the access key, secret key and region name to bridge a connection between the S3 bucket and the EC2 (alternatively, used S3Fuse to mount the S3 bucket and sync with a local directory). Ran AWS S3 ls to list folders and files in S3 bucket. Copied all files to EC2, installed relevant packages and ran the machine learning .PY file. Results were stored and analyzed for accuracy (70%). Below is a copy of the inputs made to connect the S3 to EC2 and set up the environment for implementing the machine learning .PY files.

Transaction History

```
ec2-user:~/environment $ ssh ec2-user@^C2-35-170-57-254.compute-1.amazonaws.com
ec2-user:~/environment $ ^C
ec2-user:~/environment $ cat> awskey.pem
-----BEGIN RSA PRIVATE KEY-----
[omitted for privacy]
-----END RSA PRIVATE KEY-----
ec2-user:~/environment $ chmod 400 awskey.pem
ec2-user:~/environment $ ssh -i awskey.pem ubuntu@ec2-35-170-57-254.compute-1.amazonaws.com
=============================================================================
       __|  __|_  )
       _|  (     /   Deep Learning Base AMI (Ubuntu 18.04) Version 31.0
      ___|\___|___|
=============================================================================
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1029-aws x86_64v)
Nvidia driver version: 450.80.02
CUDA versions available: cuda-10.0 cuda-10.1 cuda-10.2 cuda-11.0
Default CUDA version is 10.0
Libraries: cuDNN, NCCL, Intel MKL-DNN
AWS Deep Learning AMI Homepage: https://aws.amazon.com/machine-learning/amis/
```

```
Developer Guide and Release Notes: https://docs.aws.amazon.com/dlami/latest/devguide/what-is-
dlami.html

Support: https://forums.aws.amazon.com/forum.jspa?forumID=263

For a fully managed experience, check out Amazon SageMaker at https://aws.amazon.com/sagemaker

When using INF1 type instances, please update regularly using the instructions at:
https://github.com/aws/aws-neuron-sdk/tree/master/release-notes

=============================================================================

 * Documentation:  https://help.ubuntu.com

 * Management:     https://landscape.canonical.com

 * Support:        https://ubuntu.com/advantage

  System information as of Sun Nov 29 23:04:22 UTC 2020

  System load:  0.04               Processes:              151

  Usage of /:   41.2% of 72.64GB   Users logged in:        0

  Memory usage: 1%                 IP address for eth0:    10.0.1.230

  Swap usage:   0%                 IP address for docker0: 172.17.0.1

 * Introducing self-healing high availability clusters in MicroK8s.

   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

 * Canonical Livepatch is available for installation.

   - Reduce system reboots and improve kernel security. Activate at:

     https://ubuntu.com/livepatch

23 packages can be updated.

0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet
connection or proxy settings

Last login: Mon Nov 16 05:12:49 2020 from 108.18.228.160

ubuntu@ip-10-0-1-230:~$ aws configure

AWS Access Key ID [None]: [omitted for privacy]

AWS Secret Access Key [None]: [omitted for privacy]

Default region name [None]: us-east-1

Default output format [None]:

ubuntu@ip-10-0-1-230:~$ aws s3 ls

2020-11-15 21:46:52 deepseaimages

ubuntu@ip-10-0-1-230:~$ cd deepseaimages

ubuntu@ip-10-0-1-230:~$ aws s3 ls s3://deepseaimages/FinalProject/

                           PRE Development/

                           PRE Fixed_Code/

                           PRE data/

                           PRE data_hold/
```

```
                           PRE data_random/
                           PRE data_subset/
                           PRE data_subset_2/
                           PRE old code/
                           PRE test_data/
2020-11-15 22:57:54     501237 160053-depth-template-16x9.pptx
2020-11-15 22:57:54       1993 Extractions_Download.py
2020-11-15 22:57:54      27194 Final Project Proposal.docx
2020-11-15 22:57:54     144519 Final Project Proposal.pdf
2020-11-15 22:57:54     298988 FinalProject_ML2_R(1).pdf
2020-11-15 22:57:54      15193 Jennifer_Crosby_final_project.docx
2020-11-15 22:57:57     865957 Jennifer_Crosby_final_project.pdf
2020-11-15 22:57:58     752840 Jennifer_Crosby_final_project.pptx
2020-11-15 22:57:57      17635 Jennifer_Crosby_final_report.docx
2020-11-15 22:57:57      82825 Jennifer_Crosby_final_report.pdf
2020-11-15 22:57:55    2884547 cnnmodel.pt
2020-11-15 22:57:54         43 create_files.py
2020-11-15 22:57:54      13134 final_fix.py
2020-11-15 22:57:54       1568 image_folder.py
2020-11-15 22:57:57       7150 train_finalproject (1).py
2020-11-15 22:57:57       7154 train_finalproject.py
2020-11-15 22:57:57       6984 train_finalproject_old.py
ubuntu@ip-10-0-1-230:~$ aws s3 cp s3://deepseaimages/FinalProject/final_fix.py .
download: s3://deepseaimages/FinalProject/final_fix.py to ./final_fix.py
ubuntu@ip-10-0-1-230:~$ df -k
Filesystem     1K-blocks     Used Available Use% Mounted on
udev            16428960        0  16428960   0% /dev
tmpfs            3288660      820   3287840   1% /run
/dev/xvda1      76171248 31414688  44740176  42% /
tmpfs           16443280        0  16443280   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs           16443280        0  16443280   0% /sys/fs/cgroup
/dev/loop0         18432    18432         0 100% /snap/amazon-ssm-agent/1566
/dev/loop1         96128    96128         0 100% /snap/core/8935
/dev/loop2        100096   100096         0 100% /snap/core/10185
/dev/loop3         56704    56704         0 100% /snap/core18/1932
/dev/loop4         33152    33152         0 100% /snap/amazon-ssm-agent/2996
tmpfs            3288656        0   3288656   0% /run/user/1000
```

```
ubuntu@ip-10-0-1-230:~$ mkdir images
ubuntu@ip-10-0-1-230:~$ cd images
ubuntu@ip-10-0-1-230:~/images$ history
    1  sudo yum update
    2  sudo update
    3  update
    4  curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
    5  unzip awscliv2.zip
    6  sudo ./aws/install
    7  aws s3 ls
    8  aws
    9  aws configure
   10  aws s3 ls
   11  aws configure
   12  aws s3 ls
   13  cd deepseaimages
   14  aws s3 ls
   15  aws s3 ls s3://deepseaimages
   16  aws s3 ls s3://deepseaimages/FinalProject
   17  aws s3 ls s3://deepseaimages/FinalProject/
   18  aws s3 cp s3://deepseaimages/FinalProject/final_fix.py
   19  aws s3 cp s3://deepseaimages/FinalProject/final_fix.py .
   20  df -k
   21  mkdir images
   22  cd images
   23  history
ubuntu@ip-10-0-1-230:~/images$ aws s3 cp --recursive s3://deepseaimages/FinalProject/data_subset
ubuntu@ip-10-0-1-230:~/images$
ubuntu@ip-10-0-1-230:~/images$ sudo apt install yum
ubuntu@ip-10-0-1-230:~/images$ python --version
Python 2.7.17
ubuntu@ip-10-0-1-230:~/images$ cd ..
ubuntu@ip-10-0-1-230:~$ sudo apt install python3.7-minimal
ubuntu@ip-10-0-1-230:~$ python --version
Python 2.7.17
ubuntu@ip-10-0-1-230:~$ dir
Nvidia_Cloud_EULA.pdf  README  aws  awscliv2.zip  final_fix.py  images  src  tools
```

```
ubuntu@ip-10-0-1-230:~$ cd tools

ubuntu@ip-10-0-1-230:~/tools$ dir

GPUCloudWatchMonitor

ubuntu@ip-10-0-1-230:~/tools$ cd ..

ubuntu@ip-10-0-1-230:~$ dir

Nvidia_Cloud_EULA.pdf  README  aws  awscliv2.zip  final_fix.py  images  src  tools

ubuntu@ip-10-0-1-230:~$ sudo apt-get install python3.7

ubuntu@ip-10-0-1-230:~$ python3 --version

Python 3.6.9

ubuntu@ip-10-0-1-230:~$ curl -O https://bootstrap.pypa.io/get-pip.py

Successfully installed pip-20.2.4

ubuntu@ip-10-0-1-230:~$ ls -a ~

.     .bash_history  .cache    .jupyter  .rpmdb                     .zshrc                 aws
get-pip.py  tools

..    .bash_logout   .dlamirc  .local    .ssh                       Nvidia_Cloud_EULA.pdf
awscliv2.zip  images

.aws  .bashrc        .gnupg    .profile  .sudo_as_admin_successful  README
final_fix.py  src

ubuntu@ip-10-0-1-230:~$ export PATH=LOCAL_PATH:$PATH
```