

Bài 31 - Amazon Virtual Machine Deep Learning

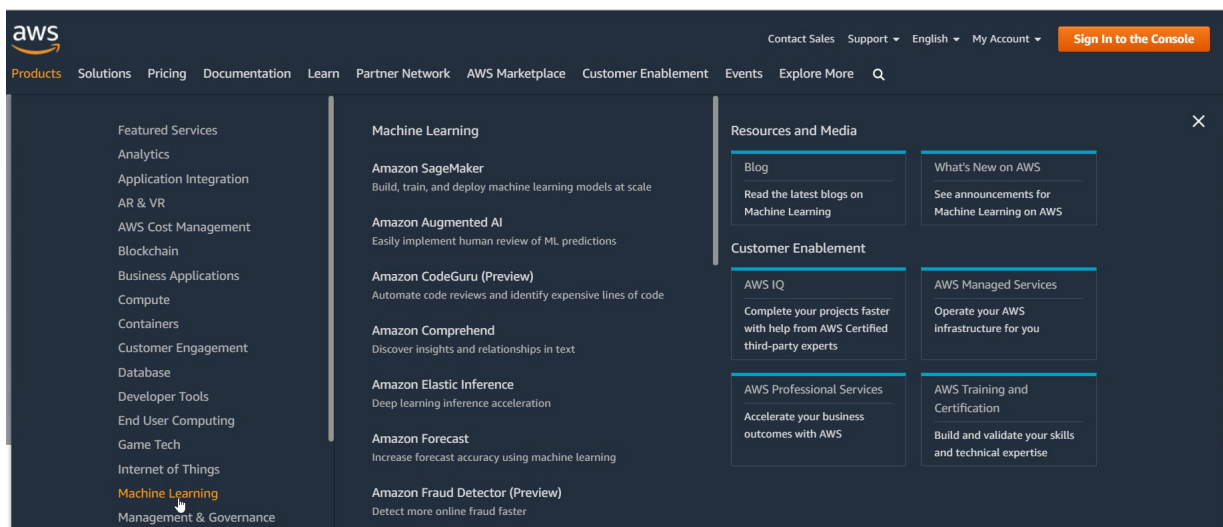
03 Apr 2020 - phamdinhhkhanh

Menu

- 1. Amazon web services
 - 1.1. Có nên đầu tư Deep Learning Computer
 - 1.2. Dịch vụ EC2
- 2. Khởi tạo một EC2
 - 2.1. Lựa chọn cấu hình GPU
 - 2.2. Giá cả EC2
- 3. Truy cập Amazon Virtual Cloud
 - 3.1. Xác thực SSH
- 4. Kết nối local jupyter notebook tới VM
- 5. Download dữ liệu
 - 5.1. Lệnh download dữ liệu
 - 5.2. Cài đặt kaggle cli
- 6. Kết luận
- 7. Tài liệu tham khảo

1. Amazon web services

Nếu các bạn đã từng làm việc với bigdata chắc sẽ không còn xa lạ với các dịch vụ của Amazon Web Services. Đây là tổng hợp các giải pháp của Amazon như cơ sở dữ liệu, lưu trữ, tính toán, giải pháp AI, machine learning, bảo mật, networking, phân tích dữ liệu, triển khai web, app, cho thuê máy chủ,.... Hệ sinh thái Amazon Web Services là một hệ sinh thái rất rộng lớn, bao gồm nhiều dịch vụ khác nhau và được cộng đồng công nghệ sử dụng rộng rãi. Những dịch vụ nằm trong hệ sinh thái đã giúp các công ty công nghệ triển khai giải pháp của mình trên cloud với chi phí thấp hơn so với việc đầu tư hạ tầng vật lý. Bạn có thể theo dõi các dịch vụ của Amazon web services tại Amazon web Services (<https://aws.amazon.com/>):



Hình 1: Các dịch vụ của Amazon Web Services.

Top

Dịch vụ AWS có hệ thống máy chủ phân bố trên toàn cầu. Do đó bạn có thể khởi tạo máy chủ của mình nằm ở nhiều nơi trên thế giới. Tuy nhiên để người dùng truy cập nhanh hơn tới hệ thống của bạn thì nên lựa chọn region gần với quốc gia của bạn. Ở Việt Nam mình thường chọn region tại Singapo.

1.1. Có nên đầu tư Deep Learning Computer

Có nhiều bạn đặt ra câu hỏi rằng có nên đầu tư siêu máy tính hỗ trợ GPU hay thuê máy chủ Deep Learning Computing để huấn luyện các mô hình deep learning?

Mình nghĩ rất khó để đưa ra một câu trả lời chính xác phương án nào hợp lý hơn. Theo quan điểm cá nhân thì mình cho rằng đầu tư siêu máy tính có những điểm bất lợi sau:

- Gia tăng chi phí cố định: Công ty của bạn phải bỏ ra một chi phí cố định rất lớn ban đầu thay vì khấu hao từng phần khi sử dụng dịch vụ. Theo tính toán của mình thì đầu tư một máy chủ giá 10 nghìn USD sử dụng trong 5 năm thì một ngày bạn sẽ tiêu tốn khoảng: 5.4 USD/ngày. Chi phí này có thể cao hơn so với sử dụng dịch vụ máy chủ deep learning.
- Out of date công nghệ: Các hãng như NVIDIA, Intel liên tục sản xuất ra những chiếc GPU mạnh mẽ hơn. Mình nhớ vào thời điểm 2018, dòng GPU Geforce RTX 2080 Ti được đánh giá khá mạnh để huấn luyện mô hình deep learning. Nhưng hiện tại nó đã bị bỏ xa bởi các dòng mới hơn như Tesla K80 (dòng GPU default của google colab), Tesla V100.
- Nâng cấp cấu hình khó khăn: Một máy tính thường bị giới hạn các thành phần như số khay cắm RAM, GPU. Do đó việc nâng cấp lên cấu hình cao hơn sẽ gặp hạn chế. Thuê máy chủ deep learning sẽ có sẵn một số cấu hình máy phù hợp với nhu cầu sử dụng. Ngoài ra có thể điều chỉnh được tăng giảm RAM, dung lượng ổ cứng.
- Cấu hình hay lỗi: Để sử dụng được các frame work deep learning sẽ cần cài đặt CUDA, cuDNN,.... Những thư viện này thường hay bị lỗi nếu chưa có kinh nghiệm cài đặt. Trong khi đó các máy chủ deep learning sẽ được cung cấp sẵn một số môi trường như tensorflow, pytorch, mxnet.

Để đăng kí dịch vụ của aws các bạn truy cập vào link: Amazon Signin (<https://signin.aws.amazon.com/signin>)

Bạn sẽ cần mở một thẻ tín dụng quốc tế để đăng kí dịch vụ của AWS.

1.2. Dịch vụ EC2

EC2 (elastic computing) là một trong những dịch vụ đầu tiên của Amazon Cloud Computing. Platform này cho phép bạn triển khai các ứng dụng trên các máy chủ cloud được đặt ở nhiều nơi khác nhau trên toàn cầu.

EC2 có nhiều cấu hình máy mặc định phù hợp với các mục đích khác nhau như: làm server, tính toán, lưu trữ, deep learning, gaming và sử dụng cho các mục đích chung. Đồng thời bạn cũng có thể cấu hình máy cloud cho riêng mình dựa trên cấu hình máy mặc định như mở rộng RAM, ổ cứng,... Đối với cấu hình máy có 4 Thông số chính mà chúng ta quan tâm:

- CPU: Số lượng thread CPUs, càng nhiều thì xử lý đa luồng càng nhanh chóng.
- GPU: Có 2 dòng GPU chính là GPU card đồ họa phục vụ cho các máy Gaming và GPU Computing hỗ trợ các model deep learning. Với mục đích là xây dựng mô hình deep learning, bạn nên lựa chọn các dòng GPU phù hợp với deep learning.
- RAM: Dung lượng RAM cho máy.
- Volume: Dung lượng ổ cứng. Thông thường các bộ dữ liệu bigdata thường rất lớn. Dữ liệu video, âm thanh và ảnh thường lớn hơn dữ liệu text. Một bộ dữ liệu có thể tốn vài chục GB

Top

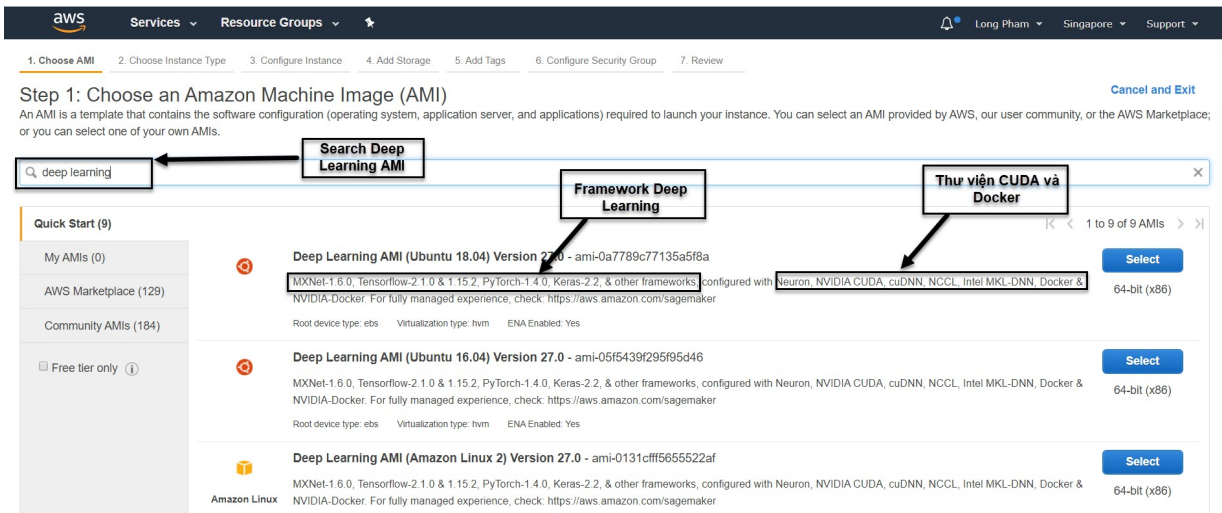
là bình thường. Do đó bạn cần phải cấu hình dung lượng ổ cứng sao cho có thể chứa được các bộ dữ liệu này.

- Region: Là vị trí đặt máy chủ vật lý. Bạn nên lựa chọn vị trí máy chủ gần với quốc gia mình để không bị các sự cố đường truyền.

2. Khởi tạo một EC2

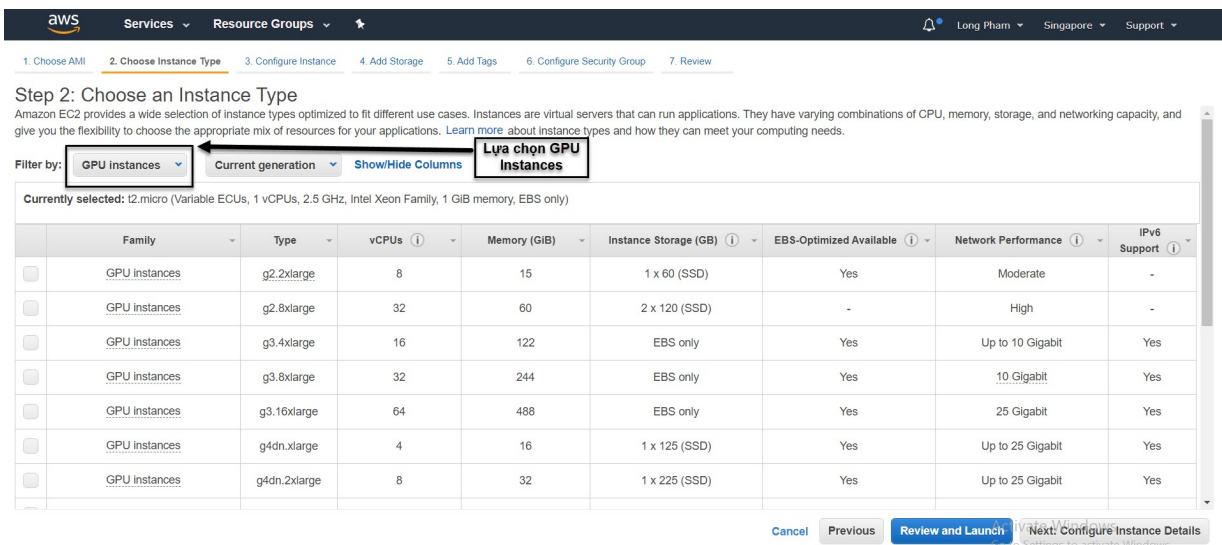
Để khởi tạo máy chủ EC2 (chúng ta sẽ gọi là instance) các bạn vào trang chủ của EC2 theo phân bố ở từng khu vực. Chẳng hạn tôi chọn ap-southeast-1/ec2 (<https://ap-southeast-1.console.aws.amazon.com/ec2/v2/home>) để khởi tạo một instance tại singapo.

Một màn hình cấu hình hiện ra cho phép chúng ta lựa chọn cấu hình máy theo 7 bước:



Hình 2: Màn hình cấu hình máy. Khi lựa chọn cấu hình máy tại bước đầu tiên chúng ta nên search “deep learning” để tìm các cấu hình máy đã cài sẵn môi trường cho các thư viện deep learning, các thư viện tính toán deep learning như CUDA, cuDNN, NCCL, Intel MKL-DNN, docker và hỗ trợ GPU. Nếu không lựa chọn đúng cấu hình máy, các bạn sẽ không có GPU và các thư viện deep learning để huấn luyện mô hình.

2.1. Lựa chọn cấu hình GPU



Hình 3: Các cấu hình GPU

Amazon đã phân chia các phân hệ máy của mình phù hợp với từng mục đích sử dụng. **GPU** Family là họ các phân hệ máy tính mà bạn chọn lựa, được chia thành các loại như storage, computing, general, GPU,....

Với mục đích là huấn luyện model Deep Learning thì mình sẽ lựa chọn Family là GPU instances. Lưu ý là trong phân hệ GPU thì có cả gaming và deep learning và chia thành 4 dòng máy chính là: g2, g3, g4dn, p2, p3. Đuôi suffix của các máy đều là large tức máy có cấu hình cao. Bên dưới các dòng máy này là medium (trung bình) và small (nhỏ).

Bạn cần phải hiểu từng dòng máy nào sử dụng GPU gì, phù hợp với deep learning hay gaming, sau đó mới quyết định lựa chọn để phù hợp với mục đích sử dụng. Bên dưới mình sẽ lý giải sơ qua các dòng máy:

- p2 : Sử dụng dòng GPU NVIDIA Tesla K80 hỗ trợ các xử lý và tính toán Machine Learning, Deep Learning. Giá từ 4-6 USD/h.
- p3 : Sử dụng dòng GPU NVIDIA Tesla V100 hỗ trợ các xử lý và tính toán Machine learning và Deep Learning. Giá trong khoảng từ 6-8 USD/h.

Về khả năng xử lý tính toán thì theo trang microway (<https://www.microway.com/knowledge-center-articles/comparison-of-nvidia-geforce-gpus-and-nvidia-tesla-gpus/>) khối lượng xử lý của Tesla V100 đối với 64 bit floating point là gấp 4 lần Tesla K80.

NVIDIA GPU Model	Double-precision (64-bit) Floating Point Performance
GeForce GTX Titan X Maxwell	up to 0.206 TFLOPS
GeForce GTX 1080 Ti	up to 0.355 TFLOPS
GeForce Titan Xp	up to 0.380 TFLOPS
GeForce Titan V	up to 6.875 TFLOPS
GeForce RTX 2080 Ti	estimated ~0.44 TFLOPS
Titan RTX	estimated ~0.51 TFLOPS
Tesla K80	1.87+ TFLOPS
Tesla P100*	4.7 ~ 5.3 TFLOPS
Quadro GP100	5.2 TFLOPS
Tesla V100*	7 ~ 7.8 TFLOPS
Quadro GV100	7.4 TFLOPS
Quadro RTX 6000 and 8000	~ 0.5 TFLOPS
Tesla T4	estimated ~0.25 TFLOPS

Hình 4: Bảng khối lượng xử lý TFLOPS của các dòng GPU đối với 64 bit floating point.

FLOPS là gì?

Nếu bạn đọc các bài báo về các mô hình object detection thì bên cạnh chỉ tiêu là số khung hình/giây (fps - frame per seconds) thường thấy có một chỉ tiêu khác là FLOPS.

Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46		link
SSD500	COCO trainval	test-dev	46.5	-	19		link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Hình 5: Bảng bench mark các models object detection.

Bạn đã từng thắc mắc FLOPS (floating point operation per second) là gì chưa? Đây chính là chỉ số đo lường số lượng các phép toán tử (operation) cần được thực hiện trên một mạng trong 1 giây. Chỉ số này càng lớn thì mô hình càng cần nhiều các xử lý tính toán và do đó tốc độ khung hình/giây của nó càng chậm. TFLOPS còn gọi là Tera FLOPS cũng tương tự như FLOPS nhưng là đơn vị tính toán lớn hơn. 1 tera FLOPS = 1 nghìn tỷ FLOPS.

Các dòng bắt đầu bằng chữ `g` là dành cho đồ họa và gaming:

- `g2` : Hỗ trợ GPU NVIDIA GRID GPU.
- `g3` : Hỗ trợ GPU NVIDIA Tesla M60 GPUs. Mạnh hơn `g2`.
- `g4dn` : Hỗ trợ NVIDIA T4 GPU. Mạnh hơn `g3`.

Tên của một dòng máy sẽ nằm ở cột Type (xem lại hình 3) có cú pháp là `[dòng máy].[số lượng GPU]xlarge`. Ví dụ: Type là `p2.8xlarge` nghĩa là dòng máy có 8 GPU Tesla K80 và thuộc dòng máy cấu hình cao. Tất nhiên chi phí sử dụng của nó sẽ cao hơn so với các dòng `p2.xlarge` chỉ gồm 1 GPU. Ngoài ra các cột khác có ý nghĩa như sau:

- Cột `vCPUs` là số lượng threads CPU của máy. Càng lớn thì hiệu năng xử lý đa luồng càng cao.
- Cột `Memory (GiB)` là dung lượng RAM. Đối với máy deep learning bạn nên chọn RAM ít nhất là 32 GB. Trong trường hợp lỗi hết RAM (`out of memory`) bạn cần điều chỉnh giảm kích thước `batch size`.

Top

- Cột Instance Storage (GB) là dung lượng ổ cứng. Trong đó ổ cứng SSD sẽ có tốc độ cao hơn EBS nhưng nhược điểm là bạn không thể sử dụng phân bổ link hoạt động dung lượng SSD giữa các máy. Trong khi đó EBS cho phép detach, attach giữa các máy với nhau.
- Cột Network Performance là tốc độ băng thông. Một điểm mà mình rất thích trên Amazon Virtual Machine đó là tốc độ băng thông rất lớn. Một bộ dữ liệu vài chục GB bạn có thể download xong trong vài phút trên Amazon Virtual Machine. Bạn sẽ rút ngắn được rất nhiều thời gian cho khâu download và giải nén dữ liệu.

2.2. Giá cả EC2

Trước khi submit cấu hình máy, bạn có thể xem qua giá cả tại EC2 pricing on-demand (<https://aws.amazon.com/ec2/pricing/on-demand/>) để ước tính chi phí bỏ ra. Giá cả sử dụng sẽ bao gồm:

- Chi phí theo yêu cầu:

Được tính theo giờ. Thường với các máy hỗ trợ từ GPU cho deep learning các mức giá từ <5 USD/h, 10-15 USD/h và 25-35 USD/h.

Phiên bản GPU – Thế hệ hiện tại

p3.2xlarge	8	31	61 GiB	Chỉ EBS	4,234 USD / giờ
p3.8xlarge	32	97	244 GiB	Chỉ EBS	16,936 USD / giờ
p3.16xlarge	64	201	488 GiB	Chỉ EBS	33,872 USD / giờ
p2.xlarge	4	16	61 GiB	Chỉ EBS	1,718 USD / giờ
p2.8xlarge	32	97	488 GiB	Chỉ EBS	13,744 USD / giờ
p2.16xlarge	64	201	732 GiB	Chỉ EBS	27,488 USD / giờ

Hình 6: Bảng giá các dòng GPU. Mình chỉ nên chọn dòng p3.2xlarge hoặc p2.xlarge thôi cho rẻ nhé mọi người.

Ví dụ bạn sử dụng máy 2h/ngày. Budget được cấp là 2000 USD thì bạn nên lựa chọn máy có price dưới $\frac{2000}{365 \times 2} = 2.7$ USD/giờ. Nếu lựa chọn máy 10-15 USD/giờ, khả năng bị vượt ngân sách là rất cao.

Với mình thì lựa chọn dòng máy p2.xlarge là đủ.

- Chi phí truyền dữ liệu:

Ngoài ra bạn còn tốn chi phí truyền dữ liệu. Chi phí này được amazon phân chia khá nhiều trường hợp nhưng theo mình review thì hầu hết dưới 0.1 USD/GB nên cũng không đáng kể lắm so với chi phí sử dụng GPU.

- Các chi phí khác: Một số chi phí khác mà Amazon sẽ thu của bạn nhưng không đáng kể lắm như:

Chi phí sử dụng IP động: Chi phí này chỉ khoảng 0.005 USD/một IP trên h.

Chi phí dự trữ công suất: Nếu dùng không hết công suất dự trữ bạn sẽ bị tính phí. Chi phí này cũng không đáng kể.

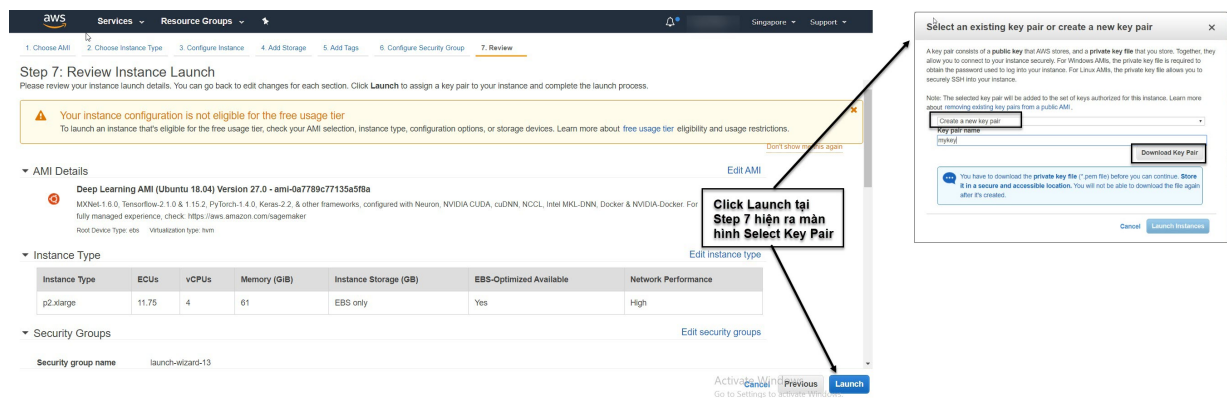
3. Truy cập Amazon Virtual Cloud ^{Top}

3.1. Xác thực SSH

Để truy cập và làm việc với VM chúng ta sẽ thông qua phương thức SSH.

SSH là Secure Shell, là một giao thức điều khiển từ xa cho phép người dùng kiểm soát và chỉnh sửa server từ xa qua Internet. Dịch vụ được tạo ra nhằm thay thế cho trình Telnet vốn không có mã hóa và sử dụng kỹ thuật cryptographic để đảm bảo tất cả giao tiếp gửi tới và gửi từ server từ xa diễn ra trong tình trạng mã hóa. Nó cung cấp thuật toán để chứng thực người dùng từ xa, chuyển input từ client tới host, và trả về kết quả cho client.

Để truy cập với Virtual Cloud, tại local, bạn cần phải cài SSH đối với hệ điều hành ubuntu và linux và OpenSSH đối với window. Bạn có thể đọc hướng dẫn cài SSH tại window SSH (https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse). Sau khi cài đặt xong SSH. Tại máy local, bạn cần và khởi tạo một SSH client. Từ SSH client chúng ta sẽ sử dụng một file mã hóa .pem để truy cập tới Virtual Machine (SSH server).



Hình 7: Save file .pem xác thực securities.

Để lấy file .pem, tại bước 7 của xác thực bạn download key pair và save vào folder dưới một tên bất kỳ. Chẳng hạn mình lấy tên file là mykey.pem.

Sau đó chúng ta sẽ truy cập vào Virtual Machine và làm theo các bước sau:

Bước 1: Phân quyền admin execute cho file .pem.

```
chmod 400 mykey.pem
```

Bước 2: Truy cập vào VM

```
ssh -i "mykey.pem" ubuntu@domain
```

Sau khi truy cập thành công màn hình sẽ hiển thị như sau:

4. Kết nối local jupyter notebook tới VM

Mặc định jupyter notebook sẽ hoạt động trên port 8888. Nếu bạn muốn sử dụng giao diện code trên jupyter notebook từ SSH client để run code trên SSH server thì cần mapping port 8888 của local với port 8888 của server. Khi đó tại **bước 2** ta thay đổi lệnh thành:

```
ssh -L 8888:localhost:8888 -i "mykey.pem" ubuntu@domain
```

Tham số -L có ý nghĩa mapping là mapping port 8888 của SSH server với localhost ở local.

Lúc này để khởi động jupyter notebook, tại commandline của server bạn khởi động jupyter notebook bằng lệnh:

```
jupyter notebook
```

Một đường link truy cập jupyter notebook từ SSH client sẽ được trả về. Bạn copy đường link này và truy cập bằng internet explore. Như vậy là ta đã có thể tương tác với server thông qua jupyter notebook trên client.

5. Download dữ liệu

5.1. Lệnh download dữ liệu

Để download một tập dữ liệu bạn sử dụng lệnh wget

```
wget [sourceLink] -O [targetFolder]
```

Trong đó sourceLink là file nguồn cần download dữ liệu và targetFolder là link tới folder đích cần lưu trữ.

Ngoài ra khi bạn download các tập dữ liệu lớn về thì định dạng thường ở dạng .zip .

- Khi đó giải nén bằng cách:

```
unzip [file.zip] -d [targetFolder]
```

Trong đó file.zip là tên file zip cần giải nén và targetFolder là địa chỉ cần giải nén.

- Trường hợp định dạng là .tar thì giải nén bằng lệnh:

```
tar -xvf [file.tar] -C [targetFolder]
```

Trong đó file.tar là tên file tar cần giải nén và targetFolder địa chỉ cần giải nén. Các thành phần của argument như sau:

x: Extract - giải nén file.

v: Verbose - show quá trình giải nén file.

f: File - định dạng file giải nén.

C: Copy kết quả giải nén vào một folder cụ thể.

Với các định dạng khác như gzip , bz2 bạn xem thêm tại hướng dẫn tar commandline (<https://www.tecmint.com/18-tar-command-examples-in-linux/>), rất đầy đủ và chi tiết.

5.2. Cài đặt kaggle cli

Tại sao cần cài đặt kaggle cli?

Như chúng ta đã biết kaggle là một nguồn dữ liệu rất đa dạng và phong phú. Tại đây bạn có thể tìm được rất nhiều các bộ dữ liệu đặc thù về ảnh, âm thanh, văn bản, video,... từ các công ty có nhu cầu phân tích dữ liệu. Trên tất cả các nền tảng data competition thì mình thấy kaggle là nơi có cộng đồng đông đảo nhất. Tại kaggle mình học được rất nhiều code từ các master và các ý tưởng phân tích hay. Tuy nhiên mình không mấy khi tham dự các cuộc thi ở đây vì mình biết rằng sẽ phải dành rất nhiều thời gian cho nó trong khi công việc của mình thì khá bận.

Top

Tại sao kaggle lại có nhiều cuộc thi như vậy? Mình nghĩ lý do chính là vì chi phí để xây dựng một đội data scientist in house ở nước ngoài khá tốn kém. Do đó các công ty thường lựa chọn tổ chức cuộc thi trên kaggle. Nhược điểm là dữ liệu của họ sẽ bị public nhưng ưu điểm là bài toán của họ sẽ được giải quyết bởi những master hàng đầu về data science và chắc chắn đó là những giải pháp không tồi, thậm chí là tốt hơn nhiều so với build một đội data scientist in house. Ngay cả google cũng thường xuyên tổ chức các cuộc thi để tìm kiếm giải pháp trên kaggle.

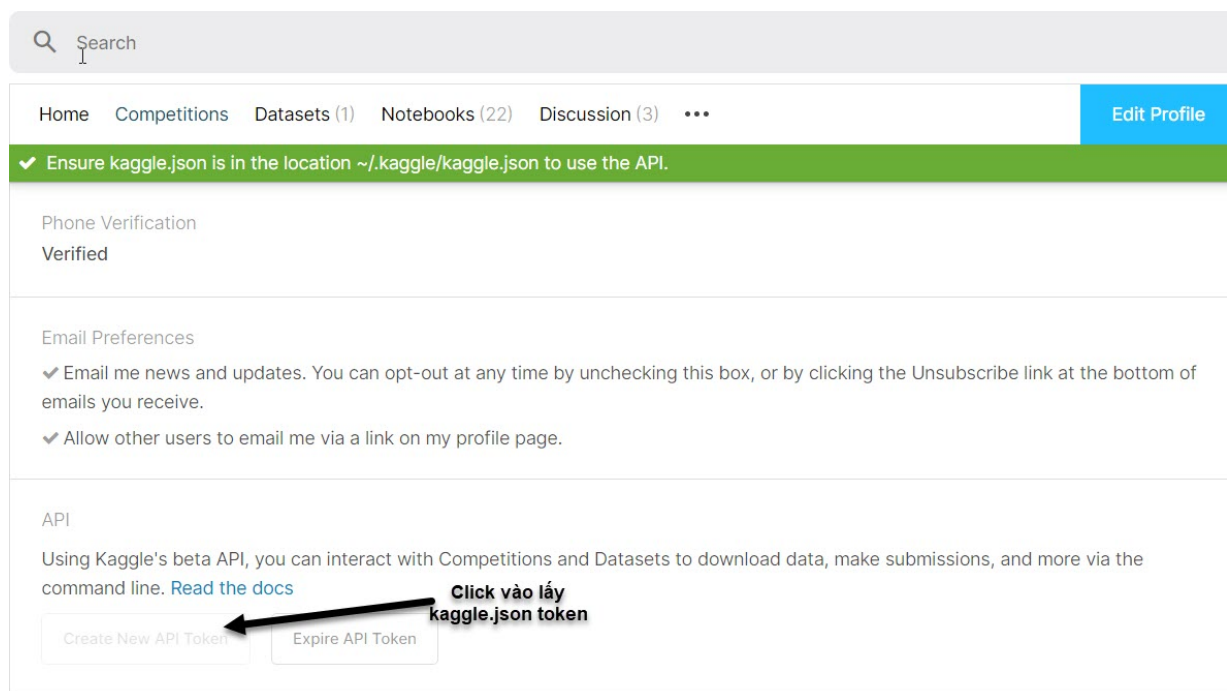
Để download nguồn dữ liệu trên kaggle, bạn phải cài đặt kaggle và kaggle cli như sau:

Bước 1: Tại SSH server bạn cài đặt các packages `kaggle`, `kaggle-cli` như sau:

```
pip3 install kaggle
```

```
pip3 install kaggle-cli
```

Bước 2: Lấy token xác thực tài khoản kaggle của bạn. Bạn cần truy cập vào kaggle account để lấy file `kaggle.json` token. Sau khi đăng nhập tài khoản kaggle, bạn vào mục `my account` sau đó tìm đến và click vào `export token` như màn hình bên dưới để lưu `kaggle.json` file về máy.



Hình 8: Lấy file xác thực token truy cập kaggle `kaggle.json` tại https://www.kaggle.com/your_username/account.

Bước 3: Upload file `kaggle.json` lên google drive hoặc dropbox. Sau đó bật chế độ share và copy link share. Mục đích chính là có thể download file từ SSH server.

Bước 4: Tại SSH server bạn download file về bằng lệnh `wget`.

```
wget [link kaggle.json share from dropbox] -O [/kaggle/kaggle.json]
```

Trong đó địa chỉ `/.kaggle/kaggle.json` là nơi mà bạn cài đặt file `kaggle.json` trên SSH server. Thường nó sẽ nằm ở ngay `root` directory. Bạn sẽ kiểm tra `kaggle.json` đã được đặt đúng vị trí chưa bằng lệnh sau:

```
ubuntu@ip-your-DNS:~$ sudo su
```

```
root@ip--your-DNS:/home/ubuntu# cd ../../
```

```
root@ip--your-DNS:/# cd kaggle
```

```
root@ip--your-DNS:/kaggle# ls
```

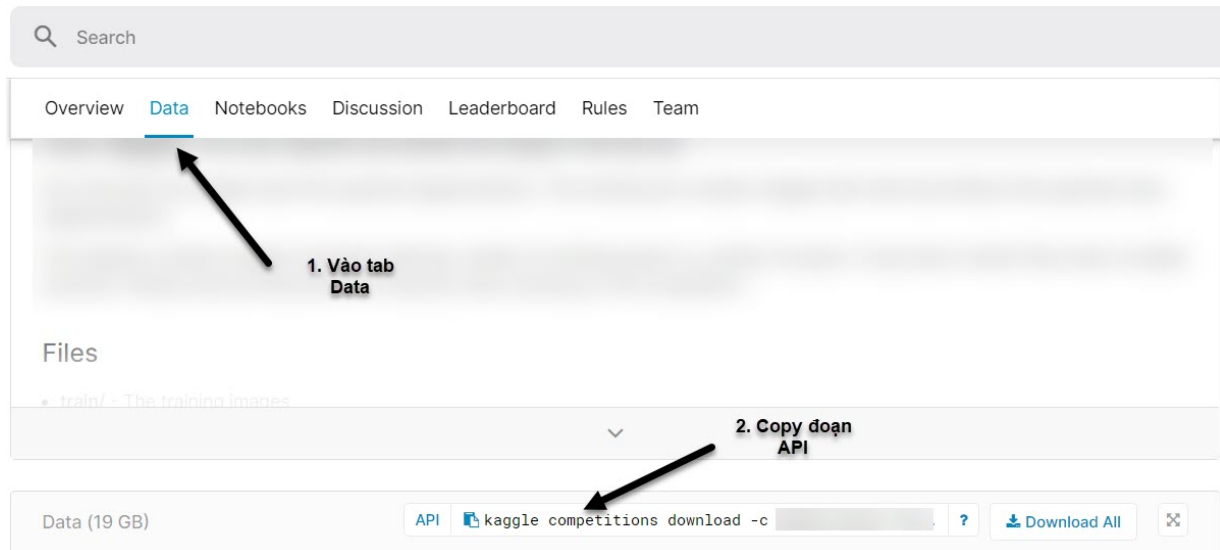
Top

kaggle.json

Nếu đã có file kaggle.json thì khi đó kaggle đã có thể xác thực token và cho phép bạn download data của họ. Để download một bộ dữ liệu bất kì bạn gõ:

```
kaggle competitions download -c [datasetname]
```

Muốn lấy data từ một cuộc thi bất kì bạn vào cuộc thi đó và lựa chọn Data > API



Hình 9: Cách lấy câu lệnh clone data từ kaggle.

6. Kết luận

Như vậy qua bài viết này mình đã giới thiệu đến các bạn các kiến thức:

- Cách thức lựa chọn Amazon Virtual Machine cho mô hình deep learning.
- Quá trình khởi tạo một Amazon Virtual Machine.
- Kết nối Jupyter notebook
- Download dữ liệu từ kaggle

Theo xu hướng, các công ty sẽ dần chuyển sang sử dụng dịch vụ cloud cho các tác vụ xây dựng mô hình Deep Learning. Do đó làm quen với công nghệ Cloud Computing sẽ giúp bạn nâng cao năng suất lao động và không bị lạc hậu. Các nhà tuyển dụng AI cũng đánh giá cao những ứng viên có kinh nghiệm về cloud computing nên đây sẽ là một điểm cộng trong kinh nghiệm làm việc của bạn.

Hi vọng rằng qua bài chia sẻ của mình, bạn sẽ có thêm những kinh nghiệm mới về các dịch vụ của Amazon Web Services.

7. Tài liệu tham khảo

1. Amazon Documentation (https://docs.aws.amazon.com/ec2/index.html?nc2=h_ql_doc_ec2).
2. Amazon Web Services Course - Coursera (<https://www.coursera.org/aws>)
3. Getting Started with AWS Machine Learning (<https://www.coursera.org/learn/aws-machine-learning>)
4. AWS Fundamentals: Building Serverless Applications (<https://www.coursera.org/learn/aws-fundamentals-building-serverless-applications>)

Top

5. AWS Training - để lấy chứng chỉ (<https://aws.amazon.com/training/>)

Top