

# Machine Learning Engineer Course

## Day 24

---

- Running the published implementation -



DIVE INTO CODE

Thursday September 30, 2021  
DIOP Mouhamed



# Agenda

---

- 1 Check-in**
- 2 Introduction**
- 3 Today's Objective**
- 4 Today's Task**
- 5 AWS**
- 6 Sample code**
- 7 To do by next class**
- 8 Check-out**



# Check-in

---

**3 minutes** Please post the following point to Zoom chat.

**Q. What did you learn in the previous week?**  
(Anything is fine.)



# Objectives of this project

---

**Read the published source code and  
Modify it according to the DIVER assignment and try to  
run it.**



# Today's Task

---

- Environment : GoogleColab
  - Models used : Faster R-CNN, YOLOv3
  - Dataset used : Simpson data
- 
- Let's run through the introductory notebooks that will be distributed and try to use colab together!



# About this slide

---

**This is the end of the explanation of Sprint.**

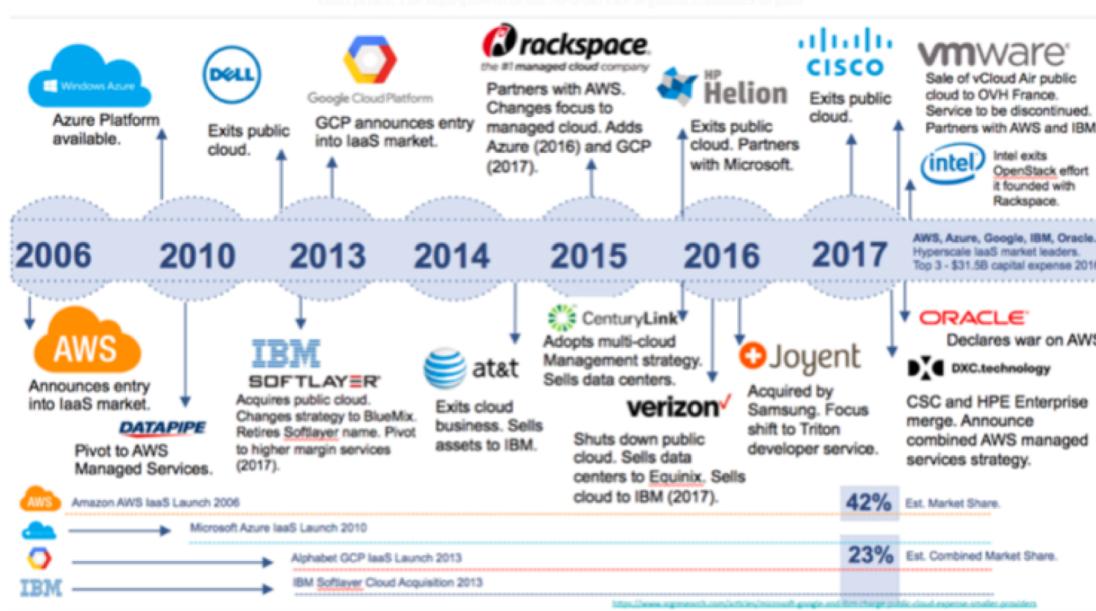
**From this page onwards, we will be discussing AWS.  
Also, this is very confusing, please do not use AWS for  
this Sprint assignment.**

**(Attention! The work we are going to proceed with will  
rent an instance of AWS; you will need to register your  
credit card information to use AWS.**

**There is also a rental fee of about 100 yen per hour. "If  
you don't want to pay a fee, please see the work.**



# About AWS



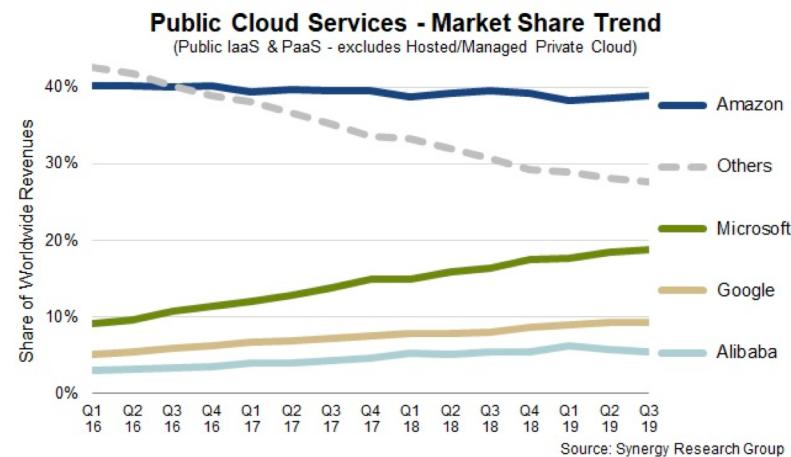
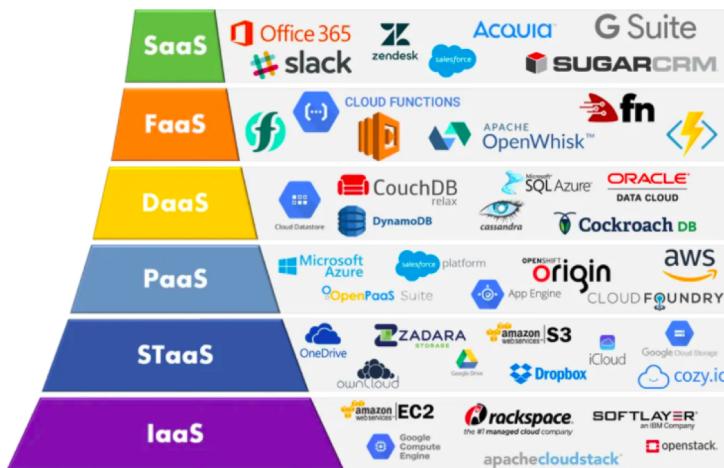
<https://medium.com/@dougpollie/there-can-be-only-one-37667ff80647>

AWS is a cloud computing service that Amazon.com (founded in 1995) has been providing since July 2006. The cloud business was initially born out of a request to solve a business problem within Amazon.com, and started with storage and virtual machine services.



# AWS service positions

As of January 2020, AWS has 191 service entries, but let's compare the market share of vendors in infrastructure as a service (IaaS) and platform as a service (PaaS).



<https://migrat.me/cloud/cloud-services-models-help-business/>

<https://www.srgresearch.com/articles/amazon-microsoft-google-and-alibaba-strengthen-their-grip-public-cloud-market>



# Using AWS

---

## In this article

### Let's use AWS virtual machines.

The service that creates and runs virtual machines (called **instances**) in the AWS cloud is called EC2 (Amazon Elastic Compute Cloud; Amazon EC2).

An instance is created as a copy of a template that records the software configuration, called an Amazon Machine Image (AMI). On the other hand, an instance has a type of hardware specification of the host computer, so different hardware types of instances can be created from the same AMI.

EC2 is classified as an IaaS service.

<https://www.quora.com/Is-Amazon-EC2-IaaS-or-PaaS>



# Renting an Instance

---

Spot instances.  
Let's rent a spot instance.

EC2 provides **three purchase options**:

- on-demand instance
  - You pay for the instances you use on a per-second basis, with no need for long-term commitments or upfront payments.
- Reserved Instance
  - You can reduce your EC2 costs by adhering to specific instance settings, including instance type and region, for a period of one to three years.
- spot instance
  - You can request unused EC2 instances to drastically reduce your EC2 costs.

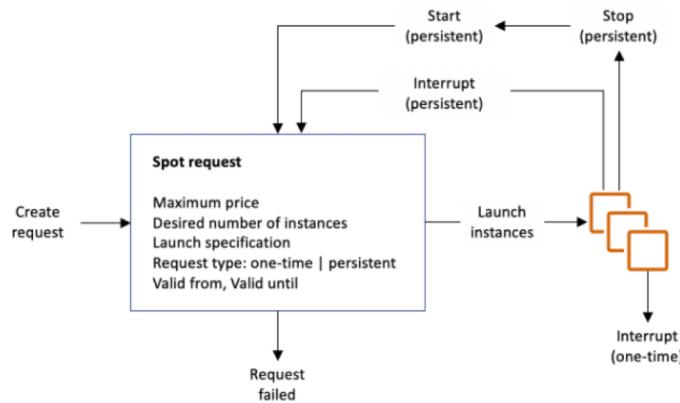
[https://docs.aws.amazon.com/ia\\_in/AWSEC2/latest/UserGuide/ec2-ug.pdf#ec2-instances-and-amis](https://docs.aws.amazon.com/ia_in/AWSEC2/latest/UserGuide/ec2-ug.pdf#ec2-instances-and-amis)



# Request Type

## Spot Instance Request Type

The one-time or persistent request type determines whether the request will be resumed when the spot instance is suspended from the Amazon EC2 side, or when the user stops the spot instance.



[https://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/spot-requests.html#using-spot-instances-request](https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/spot-requests.html#using-spot-instances-request)



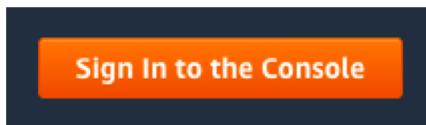
# Let's try to use AWS.

First, let's sign in.

- ① Go to AWS Home

<https://aws.amazon.com>

- ② Click the orange button in the upper right corner



- ③ When the screen looks like the one on the right, enter the email address you have registered for your account and click "Next".

- ④ Enter the password you registered and click "Sign In".

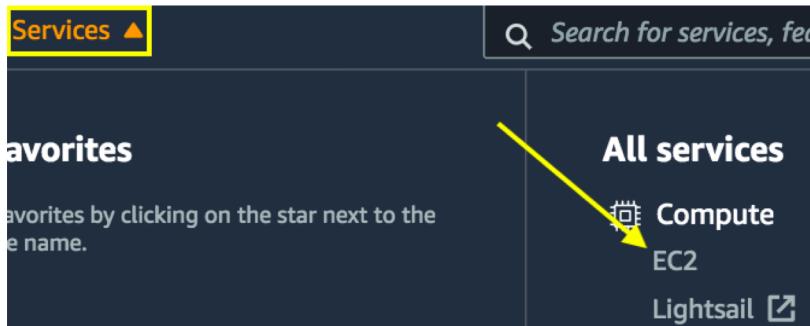
The screenshot shows the AWS sign-in interface. At the top is the AWS logo. Below it is a large 'Sign in' button. Underneath the button is a text input field with a placeholder message: 'Email address of your AWS account' followed by 'Or to sign in as an IAM user, enter your account ID or account alias instead.' Below the input field is a blue 'Next' button. At the bottom of the form are two links: 'New to AWS?' and 'Create a new AWS account'.



# Let's try to use AWS.

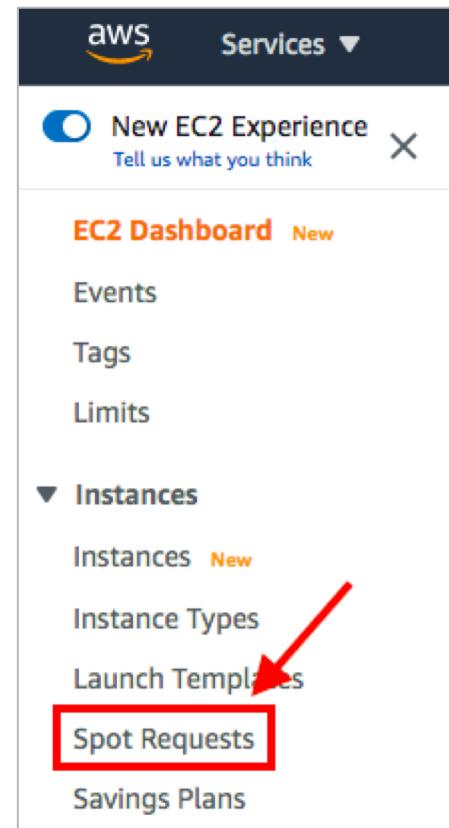
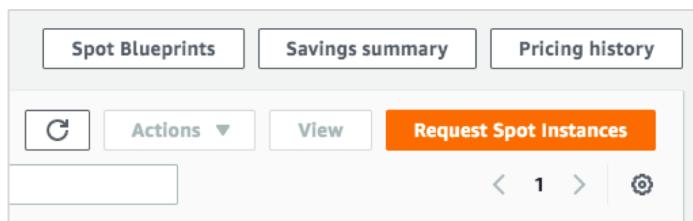
Let's use spot requests.

- ① Click "EC2" from "Services" in the upper left corner.



- ② Click on "Spot Requests" in the dashboard (see figure on the right).

- ③ Click on "Request a Spot Instance" in the upper right corner of the screen





# Let's try to use AWS.

## Select an application or task

Defined duration workloads on the far right will run continuously without interruption for 1-6 hours. The other workloads will be automatically terminated when the spot market price exceeds the bid amount.

※The rightmost one has a slightly raised bidding face.

<https://clondonaut.io/3-simple-ways-of-saving-up-to-90-of-ec2-costs/>

## Request Spot Instances

### Tell us your application or task need

To help us identify the most appropriate compute capacity for your job, select the closest match for your application or task need.

#### Load balancing workloads

Launch instances of the same size, in any Availability Zone. Good for running web services.

#### Flexible workloads

Launch instances of any size, in any Availability Zone. Good for running batch and CI/CD jobs.

#### Big data workloads

Launch instances of any size, in a single Availability Zone. Good for MapReduce jobs.

#### Defined duration workloads

Launch instances into a Spot block for 1 to 6 hours.



# Let's try to use AWS.

## Let's choose an AMI.

1. Click on the "Search AMI" button.
2. Enter "Deep Learning AMI (Ubuntu 19)" in the search field.
3. Select "Amazon AMI" from the pull-down menu.
4. From the search results, check the radio button for "Deep Learning AMI (Ubuntu) Version 19.". (Choose version 19.0 or higher.)  
**(※Be careful not to choose the Deep Learning Base AMI.)**
5. Click the "Select" button.

Amazon マシンイメージ (AMI)

The screenshot shows the AWS Lambda function configuration page. The 'Handler' field is highlighted with a red box and contains the value 'lambda\_function.lambda\_handler'. Below the table, there is a note: 'Lambda functions can't be triggered by CloudWatch Metrics. To trigger your function based on metrics, use CloudWatch Metrics triggers.' A 'Next Step' button is also visible.

名前	イメージ ID	説明	ルートデバイスタイプ	仮想化タイプ	所有者
Deep Learning AMI (Ubuntu) Version 19.3	ami-084c4bc0da31d7e12	check: <a href="https://aws.amazon.com/sagemaker">https://aws.amazon.com/sagemaker</a> With latest deep learning frameworks pre-installed: MXNet, TensorFlow, PyTorch, Keras, Chainer, Caffe/2, Theano & CNTK, configured with NVIDIA CUDA, cuDNN, NCCL & Intel MKL-DNN. For a fully managed experience, check: <a href="https://aws.amazon.com/sagemaker">https://aws.amazon.com/sagemaker</a>	ebs	hvm	898082745236
Deep Learning Base AMI (Ubuntu) Version 19.0	ami-0c491de08cf0f8ca3	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: <a href="https://aws.amazon.com/sagemaker">https://aws.amazon.com/sagemaker</a>	ebs	hvm	898082745236

閉じる 選択



# Let's try to use AWS.

## Let's choose an instance type.

1. Click the "Change Instance Type" button.
2. Search for "p2.xlarge" using cntrl + F and check the radio button (you can also search by function)
3. Click the "Select" button.

The screenshot shows the AWS Lambda console with the 'Select Instance Type' dialog open. The search bar at the top contains 'p2.xlarge'. The table below lists various instance types with their details. The 'p2.xlarge' row is highlighted with a red box and has a radio button next to it, indicating it is selected. At the bottom right of the dialog, there is a red box around the '選択' (Select) button.

インスタンスタイプ	vCPU	メモリ (GiB)	ストレージ	ネットワーク	スポット価格	スポット節約
m6g.16xlarge	64	256	EBS のみ	25 Gigabit	\$1.1737	62.95%
m6g.metal	64	256	EBS のみ	25 Gigabit	\$1.7605	44.43%
<input checked="" type="radio"/> p2.xlarge	4	61	EBS のみ	High	\$0.4626	70.00%
p2.8xlarge	32	488	EBS のみ	10 Gigabit	\$3.7008	70.00%
p2.16xlarge	64	732	EBS のみ	25 Gigabit	\$7.4016	70.00%
p3.2xlarge	8	61	EBS のみ	Up to 10 Gigabit	\$1.2582	70.00%
p3.8xlarge	32	244	EBS のみ	10 Gigabit	\$5.0328	70.00%
p3.16xlarge	64	488	EBS のみ	25 Gigabit	\$10.0656	70.00%



# Let's try to use AWS.

## Let's select a key pair.

1. Click on "Create New Key Pair".
2. Enter a key pair name (optional) and click the "Create Key Pair" button.
3. Save the key pair as it is downloaded locally.

キーペアを作成

キーペア

プライベートキーとパブリックキーの組合せで構成されるキーペアは、セキュリティ認証情報としてインスタンス接続時の ID 証明に使用されます。

名前

名前には最大 255 文字の ASCII 文字を使用できます。先頭または末尾のスペースを含めることはできません。

ファイル形式

pem  
OpenSSH で使用する場合

ppk  
PuTTY で使用する場合

キャンセル キーペアを作成



# Let's try to use AWS.

## Let's set up a fleet request.

1. Uncheck the "Apply recommendations" box.
2. **Delete all instances except "p2.xlarge"** (selected in the instance type) from the instance type.
3. The configuration is now complete. Press the "Create" button at the bottom of the page, and you'll get a green "Success" screen.

フリートリクエストの設定

推奨事項の適用

フリートリクエスト  
Amazon EC2 はこれらのインスタンスタイプからターゲット容量をリクエストします。指定したインスタンスタイプが多いほど、ターゲット容量を満たす可能性が高まります。

	インスタンスタイプ	vCPU	メモリ (GiB)	スポット価格	オンデマンド料金の削減
<input type="checkbox"/>	p2.xlarge	4	61	\$0.4626	70.00%

削除 インスタンスタイプを選択

aws サービス リソースグループ

新しい EC2 エクスペリエンス Tell us what you think × 成功 IDによるスポットリクエスト:sfr-4fedc5cb-0317-4732-a04a-7282de00585c 正常に作成されました.

EC2 ダッシュボード 新規 イベント 新規 タグ レポート 制限

▼ インスタンス インスタンス インスタンスタイプ 起動テンプレート

Actions View スポットインスタンスのリクエスト

	リクエス...	リクエスト...	インスタン...	状態	容量	ステ...	永続性	作成日
<input type="checkbox"/>	sfr-4fedc5c...	fleet	p2.xlarge	submitted	0 of 1	-	request	a few seconds ago



# Let's try to use AWS.

## Let's check the status of the spot request.

1. If the status is "active", the request has been accepted. If the status is not "active", try refreshing your browser.  
For more information on spot requests, please refer to this page.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-fleet-requests.html>

The screenshot shows the AWS EC2 console with the 'Spot Requests' section selected. A single spot request is listed:

リクエスト ID	リクエスト名	インスタンスタイプ	状態	容量	ステータス	永続性	作成日
sfr-4fedc5c...	fleet	p2.xlarge	active	1 of 1	fulfilled	request	13 minutes ago

The '状態' (Status) column for the first row is highlighted with a red box, showing the value 'active' with a green checkmark icon.



# Let's try to use AWS.

## Let's configure the instance.

1. Select an instance from the EC2 dashboard on the left side of the screen.
2. When the screen changes to the one below, **turn on the radio button** to the left of the borrowed instance ID.
3. In the instance description tab, press "**default**" in the security group.

The screenshot shows the AWS EC2 Instances dashboard at the top, displaying two instances. The second instance, with ID i-01257384fc0ddaa10, has a checked checkbox in its selection column and is highlighted with a blue border. Below the dashboard is the Instance Details page for the selected instance. The top navigation bar includes tabs for 'セキュリティ' (Security), which is underlined in orange, and other tabs like 'ネットワーキング' (Networking), 'ストレージ' (Storage), 'ステータスチェック' (Status Check), 'モニタリング' (Monitoring), and 'タグ' (Tags). The 'セキュリティ' tab displays detailed security information, including the IAM Role (empty), Owner ID (254042484370), and Launch Time (Fri Dec 18 2020 16:23:45). Under the 'セキュリティグループ' (Security Groups) section, a list shows 'sg-b8c2b4c5 (default)', which is also highlighted with a red box.

Name	インスタンス ID	インスタンスの状態	インスタンス... (Actions)	ステータスチ...
-	i-04d372da7964da33	終了済み	p2.xlarge	-
<input checked="" type="checkbox"/>	i-01257384fc0ddaa10	実行中	t2.micro	-

インスタンス: i-01257384fc0ddaa10

セキュリティ

セキュリティの詳細

IAM ロール: -

所有者 ID: 254042484370

起動時刻: Fri Dec 18 2020 16:23:45

セキュリティグループ

sg-b8c2b4c5 (default)



# Let's try to use AWS.

## Let's create a security group.

1. Click the "Create Security Group" button in the upper right corner.
2. When the screen changes to the following, enter the security group name (AI) and description (DIC)
3. Press "Add Rule" in Inbound, set the rule as follows (Outbound is OK as long as it says "All Traffic"), and press "Create" button.

The screenshot shows two steps of creating a security group:

**Step 1: Basic Details**

- セキュリティグループ名: AI (highlighted with a red box)
- 説明: DIC (highlighted with a red box)
- VPC: vpc-7f2f0818

**Step 2: Inbound Rules**

タイプ	プロトコル	ポート範囲	ソース	説明
カスタム TCP	TCP	8888	任意の... 0.0.0.0/0	
HTTP	TCP	80	任意の... 0.0.0.0/0	
SSH	TCP	22	任意の... 0.0.0.0/0	

**Buttons at the bottom:**

- ルールを追加 (Add Rule)
- Create (Create) button



# Let's try to use AWS.

## Let's prepare for SSH connection.

1. Copy the contents of "Public DNS (IPv4)" on the right side of the instance description.
2. Paste it into your local Notepad and add the following notation in red to the top of it  
**(\*\*\*is the key pair name)**

```
ssh -i ***.pem -L 8157:127.0.0.1:8888 ubuntu@ec2-13-231-193-247.ap-northeast-1.compute.amazonaws.com
```

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like 'New EC2 Experience', 'EC2 ダッシュボード', 'イベント', 'タグ', '制限', 'インスタンス' (selected), 'インスタンスタイプ', '起動テンプレート', 'スポットリクエスト', 'Savings Plans', 'リザーブドインスタンス', '専有ホスト', 'キャパシティーの予約', 'イメージ' (selected), 'AMI', and 'Elastic Block Store'. The main area shows a table of instances:

Name	インスタンス ID	インスタンスの状態	インスタンス... (Actions)	ステータスチ... (Actions)	アラームの... (Actions)	アベイラビリテ... (Actions)
-	i-0a4d372da7964da33	終了済み		p2.xlarge	-	アラーム... + ap-northeast-1a
<input checked="" type="checkbox"/>	i-01257384fc0ddaa10	実行中		t2.micro	-	アラーム... + ap-northeast-1a

Below the table, the details for instance i-01257384fc0ddaa10 are displayed. The 'Public DNS (IPv4)' field is highlighted with a red box:

詳細	セキュリティ	ネットワーキング	ストレージ	ステータスチェック	モニタリング	タグ
<b>インスタンス概要</b>						
インスタンス ID	パブリック IPv4 アドレス	プライベート IPv4 アドレス				
i-01257384fc0ddaa10	<a href="#">3.112.231.161   オープンアドレス</a>	<a href="#">172.31.40.115</a>				
インスタンスの状態	パブリック IPv4 DNS	プライベート IPv4 DNS				
実行中	<a href="#">ec2-3-112-231-161.ap-northeast-1.compute.amazonaws.com   オープンアドレス</a>	<a href="#">ip-172-31-40-115.ap-northeast-1.compute.internal</a>				



# Let's try to use AWS.

## Starting an SSH connection

1. Launch a terminal and navigate to the directory where you saved the \*\*\*.pem (key pair).
2. Run chmod 400 \*\*\*.pem (you only need to do it this time for this key pair)
3. Run the `ssh -i ***.pem -L 8157:127.0.0.1:8888 ubuntu@ec2-13-231-193-247.ap-northeast-1.compute.amazonaws.com` that you just created on Notepad.
4. When asked for yes / no, type yes. When the following is executed and the prompt changes to `ubuntu@~`, the ssh connection is complete.

```
shuji:dic_ml_ans tominagashuuji$ ssh -i a_key_pair.pem -L 8157:127.0.0.1:8888 ubuntu@ec2-13-231-193-247.ap-northeast-1.compute.amazonaws.com
=====
                         _\   _/ )  Deep Learning AMI (Ubuntu 18.04) Version 25.3
                         _\|_|_|
=====

Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1052-aws x86_64)

Please use one of the following commands to start the required environment with the framework of your choice:
for MXNet(+Keras2) with Python3 (CUDA 10.1 and Intel MKL-DNN) _____ source activate mxnet_p36
for MXNet(+Keras2) with Python2 (CUDA 10.1 and Intel MKL-DNN) _____ source activate mxnet_p27
for TensorFlow(+Keras2) with Python3 (CUDA 10.0 and Intel MKL-DNN) _____ source activate tensorflow_p36
for TensorFlow(+Keras2) with Python2 (CUDA 10.0 and Intel MKL-DNN) _____ source activate tensorflow_p27
for PyTorch with Python3 (CUDA 10.0 and Intel MKL) _____ source activate pytorch_p36
for PyTorch with Python2 (CUDA 10.0 and Intel MKL) _____ source activate pytorch_p27
for Chainer with Python2 (CUDA 10.0 and Intel iDeep) _____ source activate chainer_p27
for Chainer with Python3 (CUDA 10.0 and Intel iDeep) _____ source activate chainer_p36
for base Python2 (CUDA 10.0) _____ source activate python2
for base Python3 (CUDA 10.0) _____ source activate python3

Official Conda User Guide: https://docs.conda.io/projects/conda/en/latest/user-guide/
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
=====

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Thu Jun 18 07:16:51 UTC 2020

System load: 0.0          Processes:      121
Usage of /: 56.1% of 87.18GB  Users logged in:    0
Memory usage: 1%           IP address for ens3:  172.31.47.253
Swap usage: 0%             IP address for docker0: 172.17.0.1
```



# Let's try to use AWS.

## Let's choose a virtual environment.

1. Select any environment from the list of virtual environments (the bottom part of the EC2 logo that appears when you go up the terminal) and run it (choose the one with p36). In this case, copy source activate tensorflow\_p36 and run it.

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/\*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

```
ubuntu@ip-172-31-47-253:~$ source activate tensorflow_p36
```



# Let's try to use AWS.

## Launching the jupyter notebook

1. Type **jupyter notebook** into the command (execute)
2. Copy **the following string** token= executed in the terminal
3. Open <http://127.0.0.1:8157> (Mac / Windows)  
※ If the jupyter notebook is running locally (localhost), disconnect it.

```
(tensorflow_p36) ubuntu@ip-172-31-47-161:~$ jupyter notebook
[I 06:11:05.916 NotebookApp] Using EnvironmentKernelSpecManager...
[I 06:11:05.917 NotebookApp] Started periodic updates of the kernel list (every 3 minutes).
[I 06:11:06.137 NotebookApp] Loading IPython parallel extension
[I 06:11:06.165 NotebookApp] JupyterLab beta preview extension loaded from /home/ubuntu/anaconda3/envs/tensorlab
[I 06:11:06.165 NotebookApp] JupyterLab application directory is /home/ubuntu/anaconda3/envs/tensorflow_p36
[I 06:11:06.352 NotebookApp] [nb_conda] enabled
[I 06:11:06.355 NotebookApp] Serving notebooks from local directory: /home/ubuntu
[I 06:11:06.355 NotebookApp] 0 active kernels
[I 06:11:06.355 NotebookApp] The Jupyter Notebook is running at.
[I 06:11:06.355 NotebookApp] http://localhost:8888/?token=d735de6ecd15a9aa3829f3c619008eafc3fd1fefae7bd2c2
```



# Let's try to use AWS.

## Launching the jupyter notebook

1. Paste the previous string into the box on the screen below.
2. Press Login, and when the jupyter notebook home page changes, the connection is successful.

The screenshot shows a Jupyter Notebook login interface. At the top, there is a logo and the word "jupyter". Below that is a form with a label "Password or token:" followed by a text input field containing a series of dots (.....). To the right of the input field is a "Log in" button. A red rectangular box highlights the password input field. Below the form, there is a section titled "Token authentication is enabled" with the following text:  
If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you [enable a password](#).  
The command:  
`jupyter notebook list`  
will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:  
`Currently running servers:  
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks`  
or you can paste just the token value into the password field on this page.  
See [the documentation on how to enable a password](#) in place of token authentication, if you would like to avoid dealing with random tokens.  
Cookies are required for authenticated access to notebooks.  
**Setup a Password**  
You can also setup a password by entering your token and a new password on the fields below:  
**Token**



# Let's try to use AWS.

## Termination procedure for spot instances①

1. In the Instances section of the EC2 dashboard, make sure that the radio button for the target instance ID is checked, and from the instance status pull-down, click "Stop Instance"
1. When you see the pop-up that says "End Instance", click Press "Stop".

The screenshot shows the AWS EC2 Instances dashboard. On the left sidebar, under the 'Instances' section, the 'Instances New' item is highlighted with a red box. In the main content area, a table lists instances. A checkbox next to the first instance (with ID i-01257384fc0ddaa10) is checked and highlighted with a red box. To the right of the table, a modal dialog box titled '停止 インスタンス?' (Stop Instance?) is displayed. Inside the dialog, the instance ID 'i-01257384fc0ddaa10' is shown. Below it, a confirmation message reads: 'インスタンスを停止することを確認するには、次の停止ボタンを選択してください。' (Please select the stop button to confirm stopping the instance.) At the bottom right of the dialog is a large orange button labeled '停止' (Stop), which is also highlighted with a red box. On the far right of the dashboard, a vertical menu titled 'インスタンスの状態' (Instance Status) is open, showing options like 'インスタンスを停止' (Stop instance), 'インスタンスを開始' (Start instance), 'インスタンスを再起動' (Restart instance), and 'インスタンスを休止' (Halt instance). The 'インスタンスを停止' option is also highlighted with a red box.



# Let's try to use AWS.

## Termination procedure for spot instances②

1. In the Spot Requests section of the EC2 dashboard, make sure that the radio button for the target request ID is checked, and select "Cancel Spot Request" from the Action button pull-down. "When the "Cancel Spot Request" pop-up appears, click "Confirm".

The screenshot shows the AWS EC2 Spot Requests page. The top navigation bar includes the AWS logo, service dropdown, resource group dropdown, user shujitominaga, location Tokyo, and support dropdown. On the left, a sidebar has a feedback link and links for new EC2 experience, dashboard, events, tags, reports, limits, instances, and spot requests. The main content area shows a table for spot requests. A red box highlights the 'Actions' button, which is set to 'Spot Instances Request'. Below it, another red box highlights the 'Cancel Request' button in a modal dialog. The table lists one spot request with details: Request ID sfr-4fedc5c..., Fleet, Instance Type p2.xlarge, Status cancelled\_t, and Capacity -.

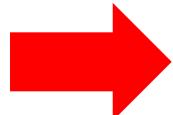


# Sprint 17 – Running published implementation

---

**Explanation about this Sprint is given but please try it on your own first.**

## Sprint 17 – Running published implementation



Please work on your own after class and submit your assignments on DIVER.

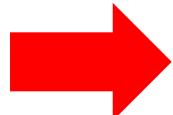


## Sprint 17 – Running published implementation

---

**A Sample Code of this Sprint is given but please try it on your own.**

### **Sprint 17 – Running published implementation**



Please work on your own after class and submit your assignments on DIVER.



# ToDo by next class

---

Next class will be Zoom : Thursday October 7, 2021 19:30 ~ 20:30

ToDo: Dataset creation

<https://diveintocode.jp/curriculums/1907>



# Check-out

---

**3 minutes** Please post the following point to Zoom chat.

**Q. Current feelings and reflections**  
(joy, anger, sorrow, anticipation, nervousness, etc.)



# Thank You For Your Attention

---

