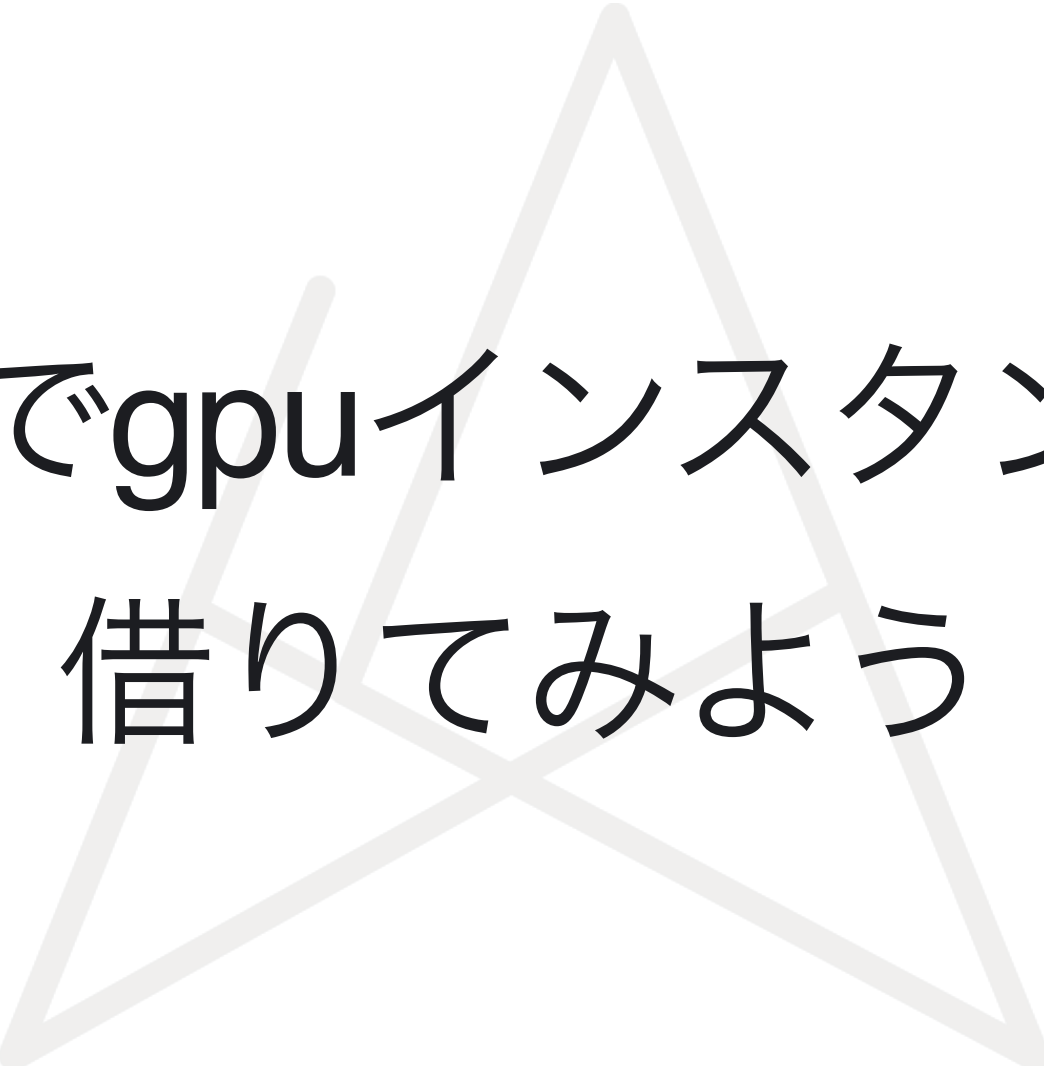


SPRINT 17



AWSでgpuインスタンスを 借りてみよう

まずはサインインしよう

①AWSホームを訪れましょう

<https://aws.amazon.com/>

②右上のオレンジのボタンを押しましょう

Sign In to the Console

③右の表示が出たら、アカウント登録したメールアドレスを入力し、**次へ**を押しましょう

④登録したパスワードを入力し、**サインイン**を押しましょう



Sign in ⓘ

Email address of your AWS account

Or to sign in as an IAM user, enter your account ID or account alias instead.

メールアドレス

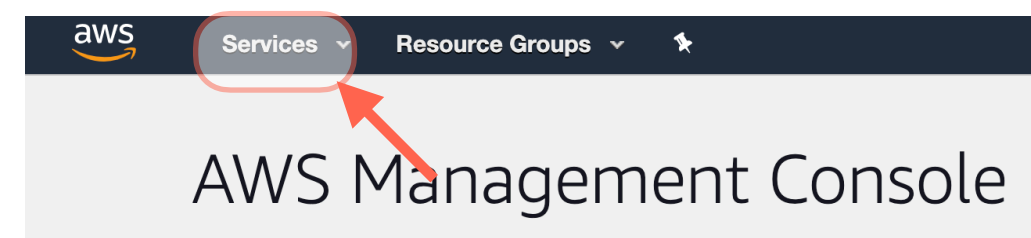
Next

— New to AWS? —

Create a new AWS account

スポットリクエストを使ってみよう

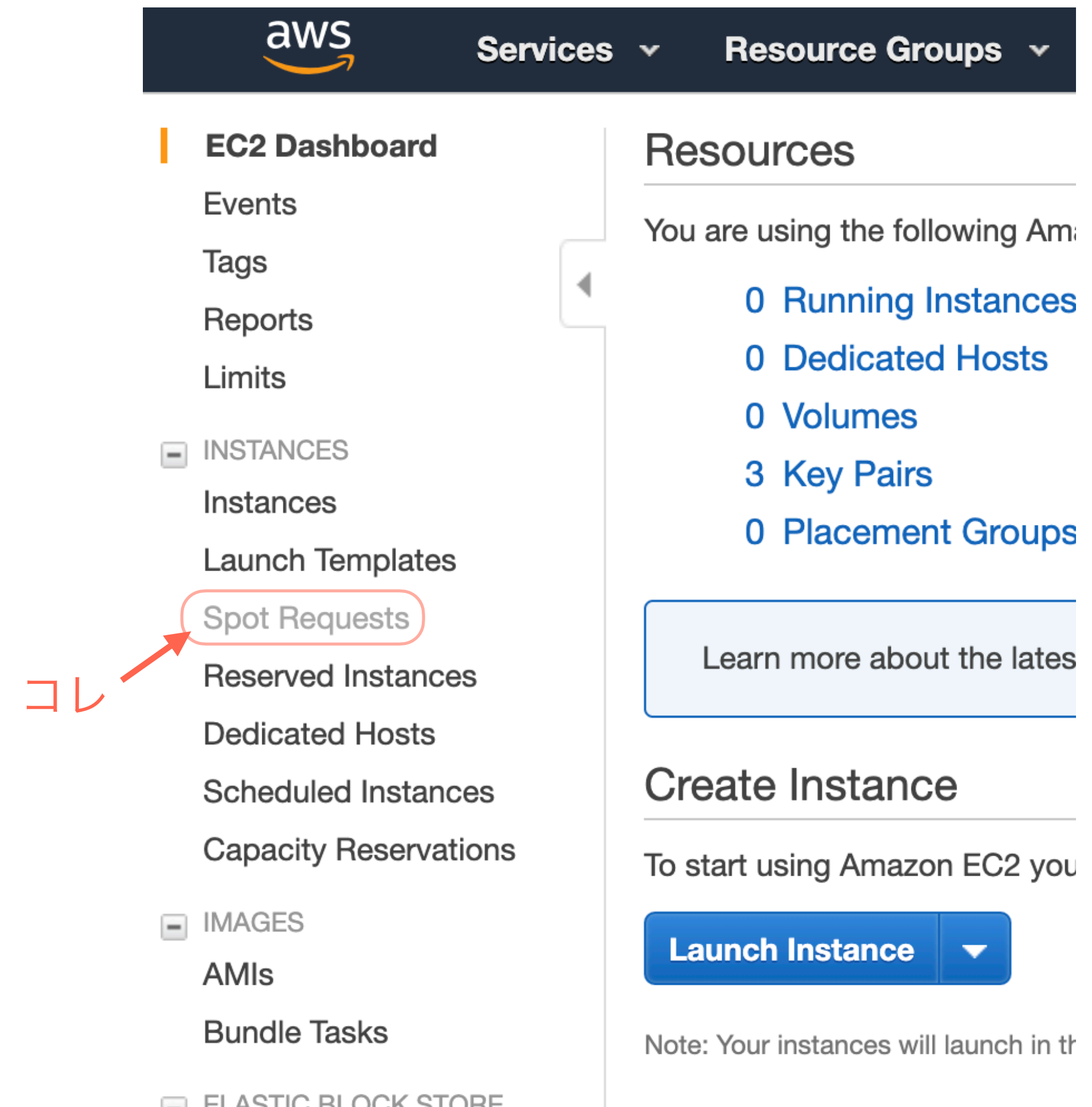
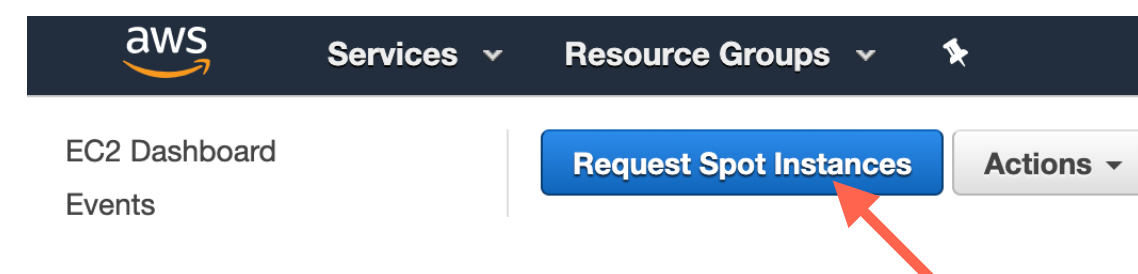
①左上のサービスボタンを押しましょう



②コンピューティングにあるEC2を選びましょう

③左のEC2ダッシュボードにあるスポットリクエストを押しましょう（右図を参考）

④上の方にあるスポットインスタンスのリクエストボタンを押しましょう



アプリケーションまたはタスクの選択

一番右にある**Defined duration workloads**は1～6時間のあいだは中断することなく継続して実行されます。他はスポット市場価格が入札額を超えたときは自動的に終了されます。

※一番右は入札額がやや高め

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

EC2 > Spot Requests > Request Spot Instances

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.

One hour

今回はコレにしてみる

AMIの選択

① 「カスタムAMIを使う」 ボタンを押す

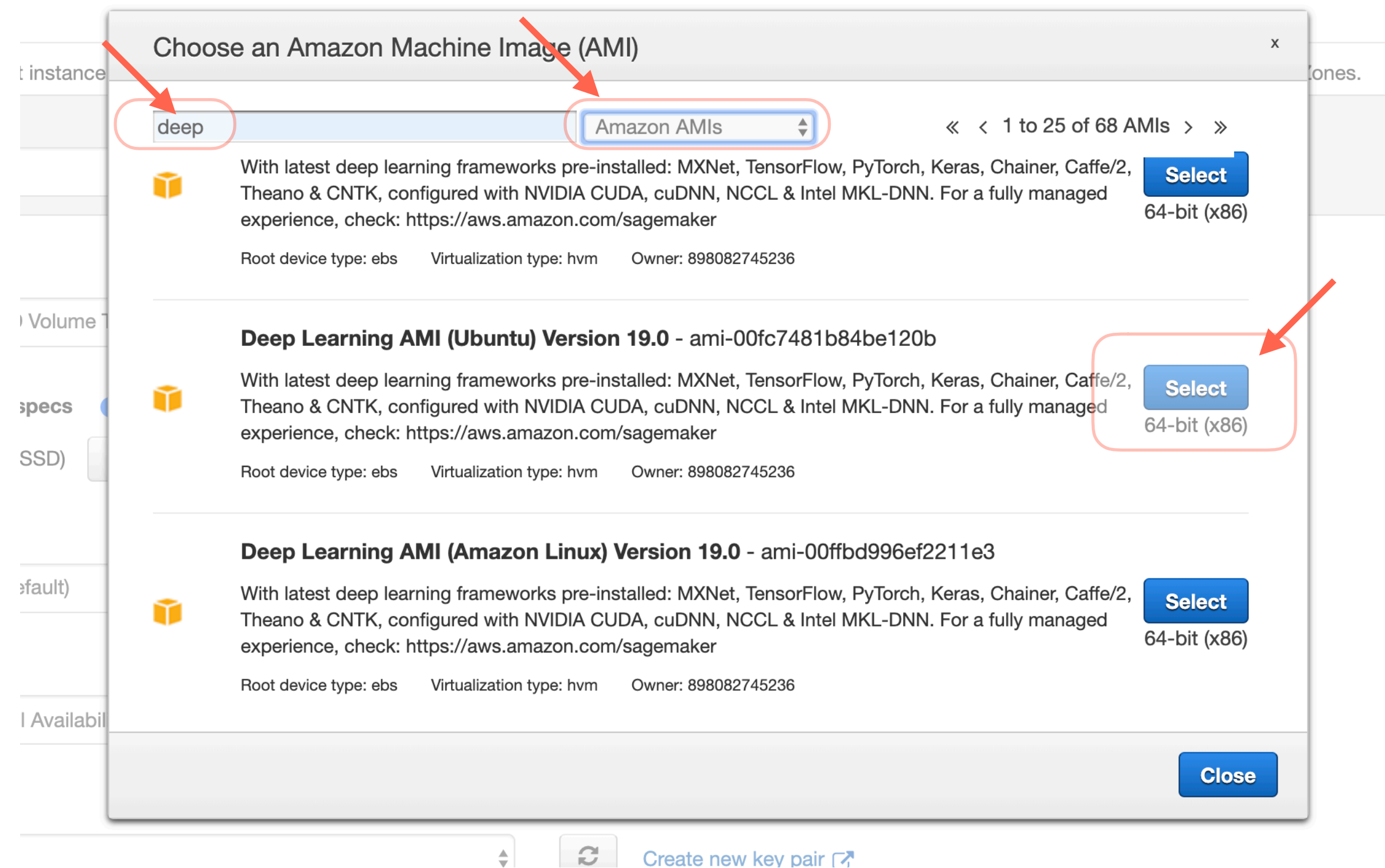
カスタム AMI を使う

② 検索窓に「deep」と入力

③ プルダウンメニューからAmazon AMIを選択して検索

④ 検索結果から「**Deep Learning AMI (Ubuntu) Version ****」を「選択」(versionは現在19.0あたり)して、「閉じる」を押す

※ 「Deep Learning Base AMI」を選ばないように注意！

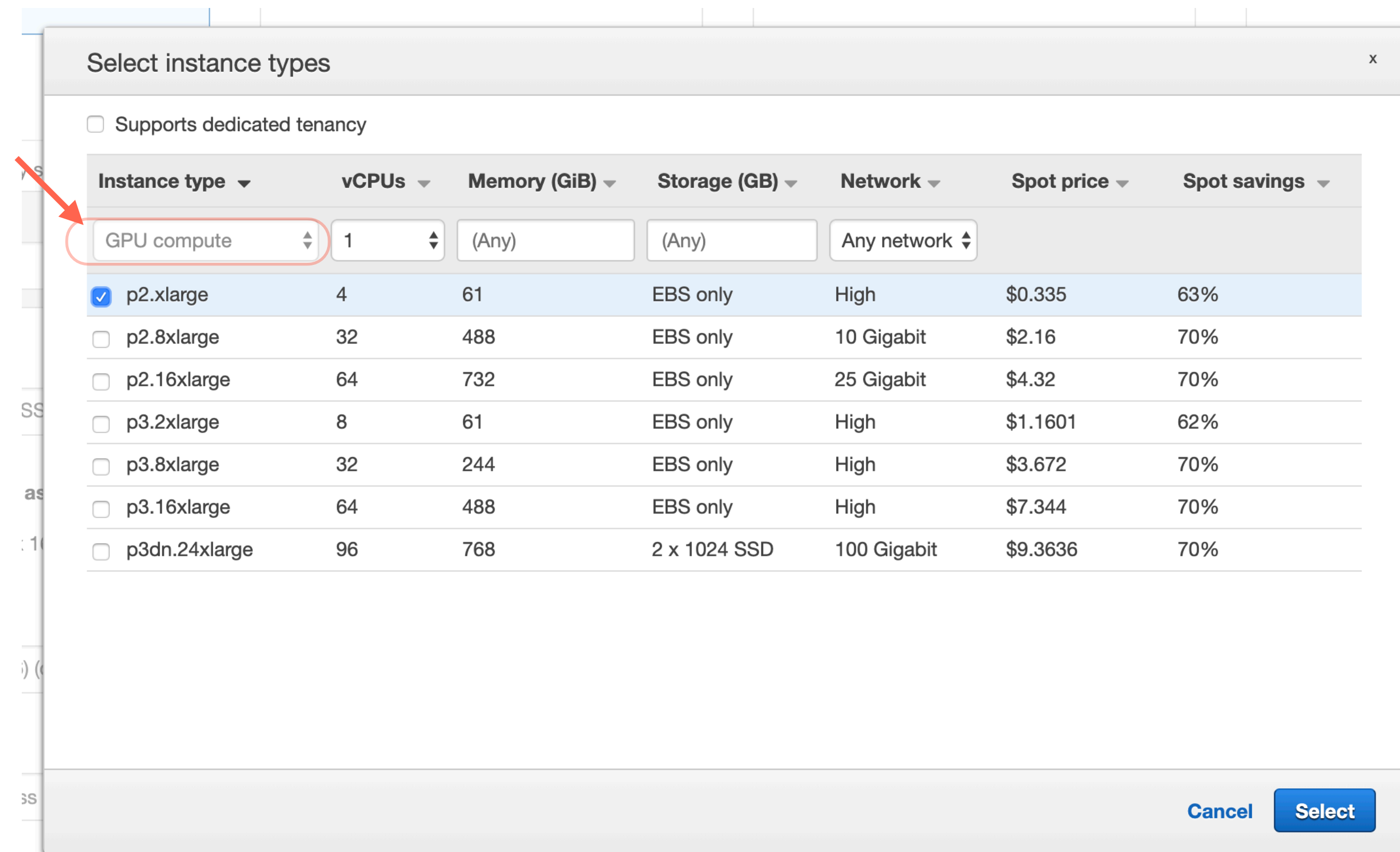


インスタンスタイプの選択

- ① 「インスタンスタイプの変更」 ボタンを押す インスタンスタイプの変更
- ② インスタンスタイプのプルダウンから「GPU Compute」を選び、検索
- ③ 「**p2.xlarge**」にチェックを入れて「選択」

※p2インスタンスのGPUはGoogle Colabと同じTesla K80

<https://aws.amazon.com/jp/ec2/instance-types/p2/>



Select instance types

☐ Supports dedicated tenancy

Instance type ▾	vCPUs ▾	Memory (GiB) ▾	Storage (GB) ▾	Network ▾	Spot price ▾	Spot savings ▾
GPU compute ▾	1 ▾	(Any)	(Any)	Any network ▾		
<input checked="" type="checkbox"/> p2.xlarge	4	61	EBS only	High	\$0.335	63%
<input type="checkbox"/> p2.8xlarge	32	488	EBS only	10 Gigabit	\$2.16	70%
<input type="checkbox"/> p2.16xlarge	64	732	EBS only	25 Gigabit	\$4.32	70%
<input type="checkbox"/> p3.2xlarge	8	61	EBS only	High	\$1.1601	62%
<input type="checkbox"/> p3.8xlarge	32	244	EBS only	High	\$3.672	70%
<input type="checkbox"/> p3.16xlarge	64	488	EBS only	High	\$7.344	70%
<input type="checkbox"/> p3dn.24xlarge	96	768	2 x 1024 SSD	100 Gigabit	\$9.3636	70%

Cancel Select

キーペアの選択

①キーペア名にある「新しいキーペア名」 ボタンを押す

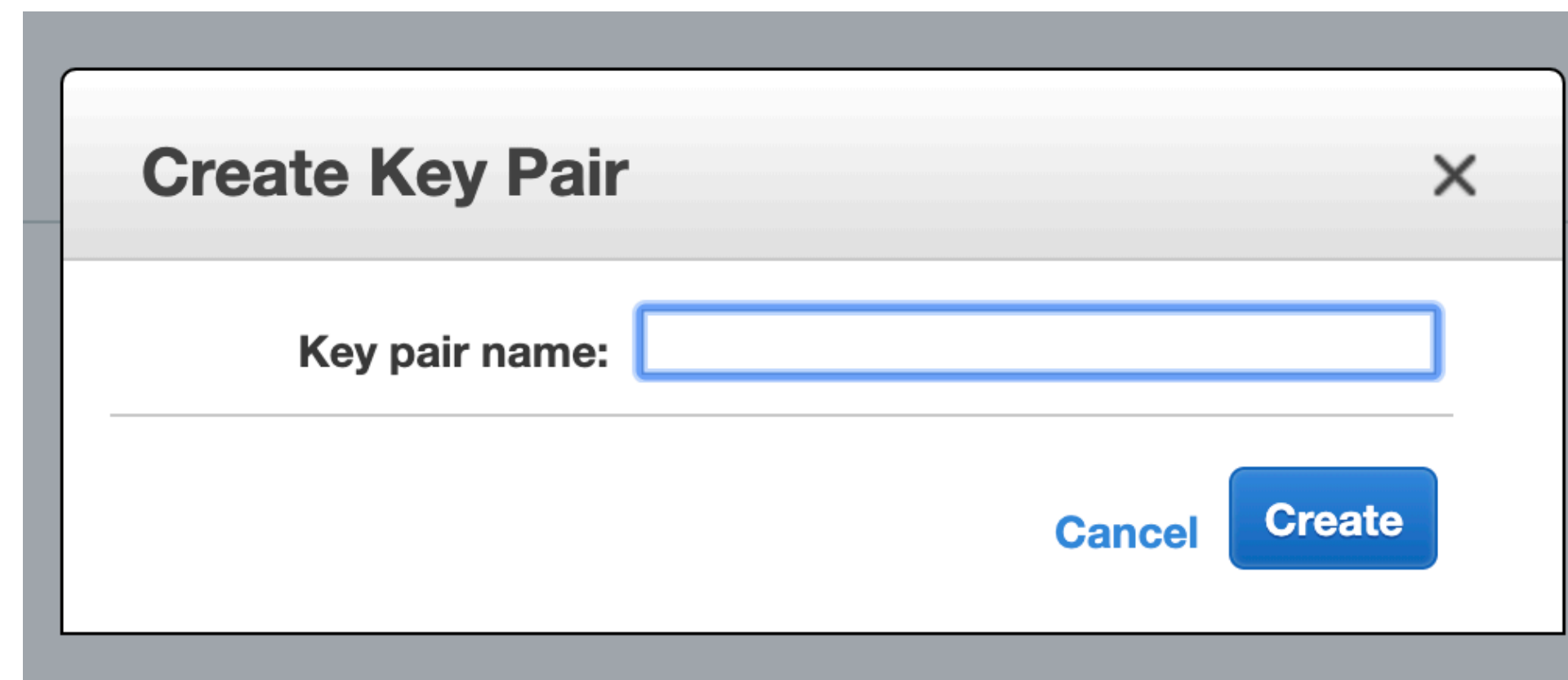


②別ウィンドウが立ち上がるので、上の方にある「キーペアの作成」 ボタンを押す



③キーペアの作成ウィンドウが起動したら、キーペア名（任意）を入力して「作成」 ボタンを押す

④ローカルにキーペアがダウンロードされるので、保存する（後で使います）



フリートリクエストの設定

- ①推奨事項の適用の **チェックボックスを外す** ☐ Apply recommendations
- ②上で選択したインスタンスタイプ「**p2.xlarge**」 **以外は削除する**
- ③これで設定終了。ページ一番下の「開始」ボタンを押して以下のような画面が立ち上がったら「OK」を押す



スポットリクエストの状態をチェック

状態が「active」になればリクエストが通っています

※スポットリクエストの状態はこちらを参考ください

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

Request Spot Instances

Actions ▾

Pricing History

Savings Summary

↺

⚙

Request type: all ▾

State: all ▾

Search by keyword

⏪

<

Viewing 1 to 3 of 3 requests

>

⏩

<input type="checkbox"/>	Request Id	Request type	Instance type	State	Capacity	Status	Persistence	Created	Max price
<input checked="" type="checkbox"/>	▼ sfr-25f11fbd-db84-...	fleet	p2.xlarge	<div><div></div>active</div>	1 of 1	fulfilled	request	2 minutes ago	\$0.9
<input type="checkbox"/>	sir-rpig5gmg	instance	p2.xlarge	<div><div></div>active</div>	i-0971be35980b...	fulfilled	one-time	2 minutes ago	\$0.9
<input type="checkbox"/>	▶ sfr-d8051523-cdb1...	fleet	p2.xlarge	<div><div></div>cancelled_...</div>	1 of 1		request	7 minutes ago	\$0.9

インスタンスの設定

- ①画面左側のEC2ダッシュボードから**インスタンス**を選択
- ②以下の画面に切り替わったら、借りたインスタンスIDの左側の**ラジオボタン**を**オン**にする
- ③インスタンスの説明の中にあるセキュリティグループの「**default**」を押す

Launch Instance ▾ Connect Actions ▾

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
<input type="checkbox"/>		i-055ffc15dd9299f62	p2.xlarge	us-east-1e	terminated		None		-
<input checked="" type="checkbox"/>		i-0971be35980bc6585	p2.xlarge	us-east-1e	running	2/2 checks ...	None	ec2-54-208-136-199.co...	54.20

Instance: **i-0971be35980bc6585** Public DNS: **ec2-54-208-136-199.compute-1.amazonaws.com**

Description Status Checks Monitoring Tags

Instance ID	i-0971be35980bc6585	Public DNS (IPv4)	ec2-54-208-136-199.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	54.208.136.199
Instance type	p2.xlarge	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-69-248.ec2.internal
Availability zone	us-east-1c	Private IPs	172.31.69.248
Security groups	default. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-48bf8231
AMI ID	Deep Learning AMI (Ubuntu) Version 19.0 (ami-00fc7481b84be120b)	Subnet ID	subnet-039f693c

セキュリティグループの作成

- ①上の「セキュリティグループの作成」ボタンを押す
- ②以下の画面が立ち上がったら、セキュリティグループ名と説明を入力
- ③インバウンドで「ルールの追加」を押し、以下のように設定し（アウトバウンドの方は**全てのトラフィック**と表示されていればOK）、「作成」ボタンを押す

Create Security Group

Create Security Group

Security group name

AI

Description

DIC

VPC

vpc-48bf8231 (default)

Security group rules:

Inbound

Outbound

Type	Protocol	Port Range	Source	Description
Custom TCP	TCP	8888	Anywhere	0.0.0.0/0, ::/0
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0

Add Rule

Cancel

Create

セキュリティグループの選択

- ①再びEC2ダッシュボードからインスタンスに戻る
- ②先ほどと同じインスタンスIDのラジオボタンをオンにする
- ③上のアクションボタンを押し、**ネットワーキング** → **セキュリティグループの変更**を選ぶ
- ④セキュリティグループ一覧が立ち上がるので、先ほど作成したセキュリティグループを選択する（それ以外のdefaultのチェックボックスは外す）

The screenshot shows the AWS Management Console interface for an EC2 instance. The instance ID is i-0971be35980bc6585. The 'Actions' menu is open, and the 'Change Security Groups' option is highlighted. A red arrow points to the radio button for the selected instance.

Name	Instance ID	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-055ffc15dd929...	us-east-1e	terminated		None		-
	i-0971be35980bc6585	us-east-1e	running	2/2 checks ...	None	ec2-54-208-136-199.co...	54.208.136.199

Instance: i-0971be35980bc6585

Description	Status Checks	Monitoring	Tags
Instance ID	i-0971be35980bc6585		
Instance state	running		
Instance type	p2.xlarge		
Elastic IPs			
Availability zone	us-east-1e		

Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Private DNS	Private IPs
ec2-54-208-136-199.compute-1.amazonaws.com	54.208.136.199	-	ip-172-31-69-248.ec2.internal	172.31.69.248

SSH接続の準備

- ①インスタンスの説明の右側にある**パブリックDNS (IPv4)**の内容をコピー
- ②ローカルのメモ帳にペーストし、その先頭に以下の赤字の表記を追加する
(***はキーペア名)

```
ssh -i ***.pem -L 8157:127.0.0.1:8888 ubuntu@ec2-54-208-136-199.compute-1.amazonaws.com
```

パブリック DNS (IPv4)	ec2-54-208-136-199.compute-1.amazonaws.com
IPv4 パブリック IP	54.208.136.199
IPv6 IP	-
プライベート DNS	ip-172-31-69-248.ec2.internal
プライベート IP	172.31.69.248
セカンダリプライベート IP	
VPC ID	vpc-48bf8231
サブネット ID	subnet-039f693c
ネットワークインターフェイス	eth0
送信元/送信先チェック	True
T2/T3 無制限	-

SSH接続の開始

- ①ターミナルを起動し、*****.pem**（キーペア）を保存したディレクトリまで移動
- ②**chmod 400 ***.pem** を実行（このキーペアに対しては今回だけ実行すれば良い）
- ③先ほどメモ帳上で作成した**ssh -i ***.pem -L 8157:127.0.0.1:8888**

ubuntu@ec2-54-208-136-199.compute-1.amazonaws.com
を実行

- ④**yes/no**を尋ねられたら、**yes**と入力する。以下が実行され、プロンプトが

ubuntu@~に変わっていればssh接続完了

```
The authenticity of host 'ec2-54-208-136-199.compute-1.amazonaws.com (54.208.136.199)' can't be established.
ECDSA key fingerprint is SHA256:zYd/pbL/32voXyQBhvwNt+JvzIy8aOUMTdTRj6yz0C4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-208-136-199.compute-1.amazonaws.com,54.208.136.199' (ECDSA) to the list of known hosts.
=====
      _| ( _|_ )
      _| ( _|_ ) /   Deep Learning AMI (Ubuntu) Version 19.0
      _| ( _|_ ) |
      _| ( _|_ ) |
      _| ( _|_ ) |
=====

Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1072-aws x86_64v)

Please use one of the following commands to start the required environment with the framework of your choice:
for MXNet(+Keras2) with Python3 (CUDA 9.0 and Intel MKL-DNN) _____ source activate mxnet_p36
for MXNet(+Keras2) with Python2 (CUDA 9.0 and Intel MKL-DNN) _____ source activate mxnet_p27
for MXNet(+Amazon Elastic Inference) with Python3 _____ source activate amazonei_mxnet_p36
for MXNet(+Amazon Elastic Inference) with Python2 _____ source activate amazonei_mxnet_p27
for TensorFlow(+Keras2) with Python3 (CUDA 9.0 and Intel MKL-DNN) _____ source activate tensorflow_p36
for TensorFlow(+Keras2) with Python2 (CUDA 9.0 and Intel MKL-DNN) _____ source activate tensorflow_p27
for TensorFlow(+Amazon Elastic Inference) with Python2 _____ source activate amazonei_tensorflow_p27
for Theano(+Keras2) with Python3 (CUDA 9.0) _____ source activate theano_p36
for Theano(+Keras2) with Python2 (CUDA 9.0) _____ source activate theano_p27
for PyTorch with Python3 (CUDA 9.2 and Intel MKL) _____ source activate pytorch_p36
for PyTorch with Python2 (CUDA 9.2 and Intel MKL) _____ source activate pytorch_p27
for CNTK(+Keras2) with Python3 (CUDA 9.0 and Intel MKL-DNN) _____ source activate cntk_p36
for CNTK(+Keras2) with Python2 (CUDA 9.0 and Intel MKL-DNN) _____ source activate cntk_p27
for Caffe2 with Python2 (CUDA 9.0) _____ source activate caffe2_p27
for Caffe with Python2 (CUDA 8.0) _____ source activate caffe_p27
for Caffe with Python3 (CUDA 8.0) _____ source activate caffe_p35
for Chainer with Python2 (CUDA 9.0 and Intel iDeep) _____ source activate chainer_p27
for Chainer with Python3 (CUDA 9.0 and Intel iDeep) _____ source activate chainer_p36
for base Python2 (CUDA 9.0) _____ source activate python2
for base Python3 (CUDA 9.0) _____ source activate python3

Official Conda User Guide: https://conda.io/docs/user-guide/index.html
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

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

93 packages can be updated.
0 updates are security updates.

*** System restart required ***

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.

=====
WARNING! Your environment specifies an invalid locale.
```

仮想環境の選択

- ①仮想環境一覧（ターミナルを上へ遡ると表示されている）から任意の環境を選択して実行（_p36がついたものを選ぶ）。今回は**source activate tensorflow_p36**をコピーして実行
- ②**pip install --upgrade pip**を求められたらこれも実行

```
sudo apt-get install language-pack-UTF-8  
or  
sudo locale-gen UTF-8
```

To see all available language packs, run:

```
apt-cache search "^language-pack-[a-z][a-z]$"
```

To disable this message for all users, run:

```
sudo touch /var/lib/cloud/instance/locale-check.skip
```

```
ubuntu@ip-172-31-69-248:~$ source activate tensorflow_p36
```

実行

jupyter notebookの立ち上げ

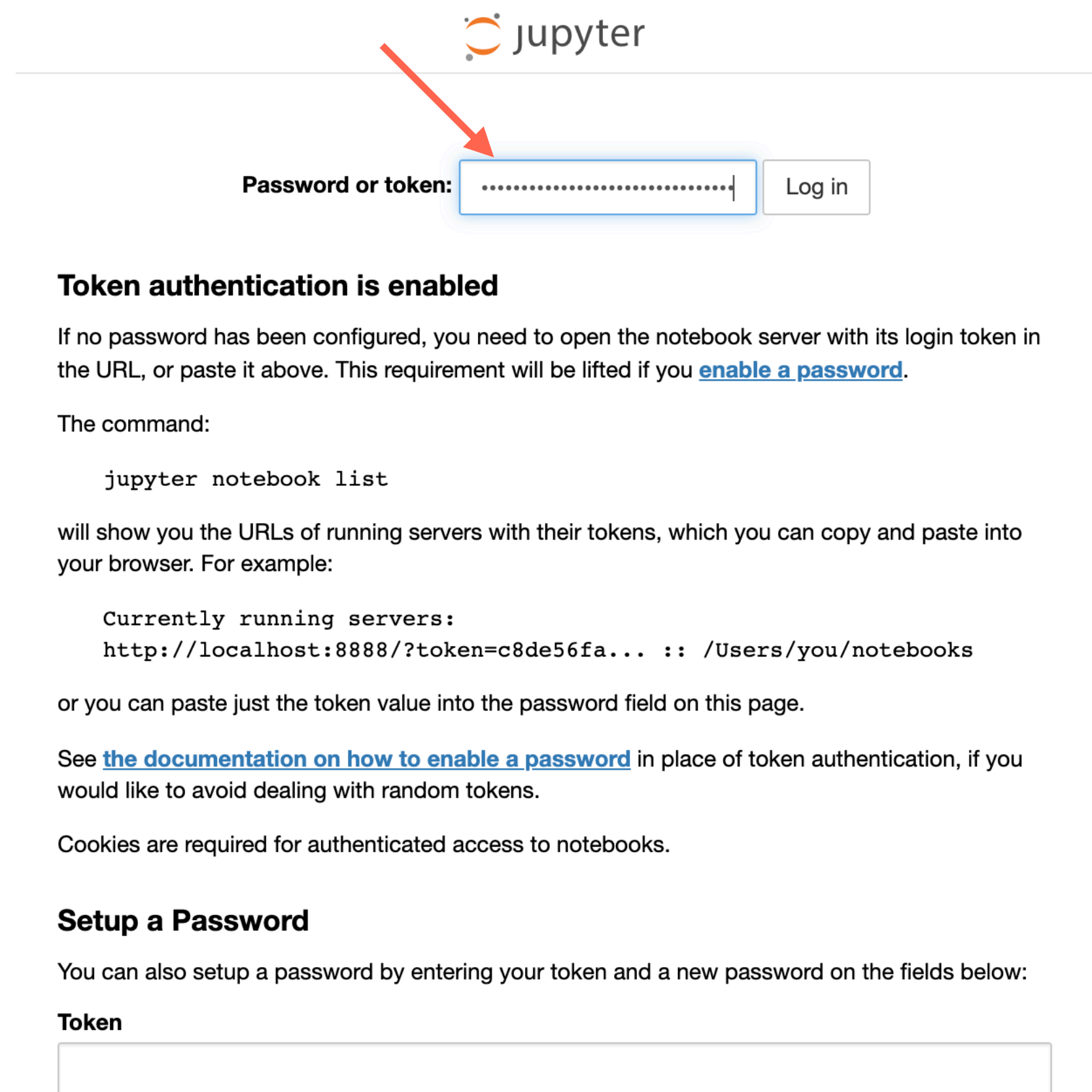
- ① **jupyter notebook**とコマンドを打つ（実行）
- ② ターミナルに表示された**token=** 以下の文字列をコピー
- ③ <http://127.0.0.1:8157>（Mac）を開く


token= 以下の文字列

```
Copy/paste this URL into your browser when you connect for the first time,  
to login with a token:  
http://localhost:8888/?token=aca9701a4a34b674bd529a6133f34e814c463ca72a34a20f&token=aca9701a4a34b674bd529a6133f34e814c463ca72a34a20f  
[ 23:31:54.406 NotebookApp] Starting initial scan of virtual environments...  
[ 23:32:50.619 NotebookApp] Found new kernels in environments: conda_cntk_p27, conda_amazonei_mxnet_p36, conda_caffe_p35, conda_theano_p36, conda.  
3. conda_caffe2_p27, conda_mxnet_p36, conda_mxnet_p27, conda_tensorflow_p36, conda_amazonei_mxnet_p27, conda_theano_p27, conda_nvthon2, conda_anai
```

jupyter notebookの立ち上げ

- ①以下の画面のボックスに先ほどの文字列をペーストする
- ②ログインを押し、jupyter notebookのホームに画面が切り替われば接続成功です

A screenshot of the Jupyter Notebook login page. At the top, there is a Jupyter logo and the word "jupyter". Below this is a horizontal line. Under the line, there is a label "Password or token:" followed by a text input field containing a series of dots. A red arrow points from the text "①以下の画面のボックスに先ほどの文字列をペーストする" to this input field. To the right of the input field is a "Log in" button. Below the login section, there is a heading "Token authentication is enabled" followed by a paragraph explaining that if no password is configured, the user needs to provide a login token in the URL or paste it into the field. It also mentions that this requirement will be lifted if the user enables a password. Below this, it says "The command:" followed by a code block containing "jupyter notebook list". It then explains that this command will show the URLs of running servers with their tokens, which can be copied and pasted into the browser. An example output is shown: "Currently running servers:" followed by a URL "http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks". It then says "or you can paste just the token value into the password field on this page." Below this, it says "See the documentation on how to enable a password in place of token authentication, if you would like to avoid dealing with random tokens." Finally, it says "Cookies are required for authenticated access to notebooks." At the bottom, there is a heading "Setup a Password" followed by a paragraph saying "You can also setup a password by entering your token and a new password on the fields below:". Below this, there is a label "Token" followed by an empty text input field.

 jupyter

Password or token:

Token authentication is enabled

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