

スポットリクエストを使ってみよう

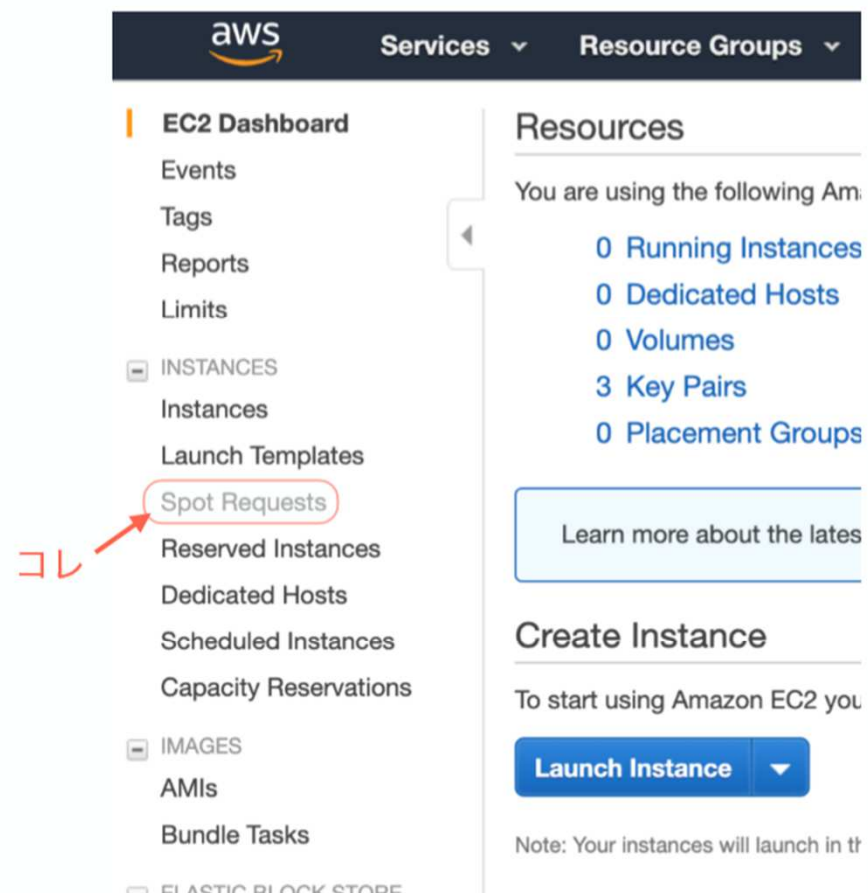
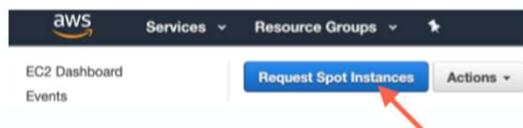
①左上のサービスボタンを押しましょう



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 

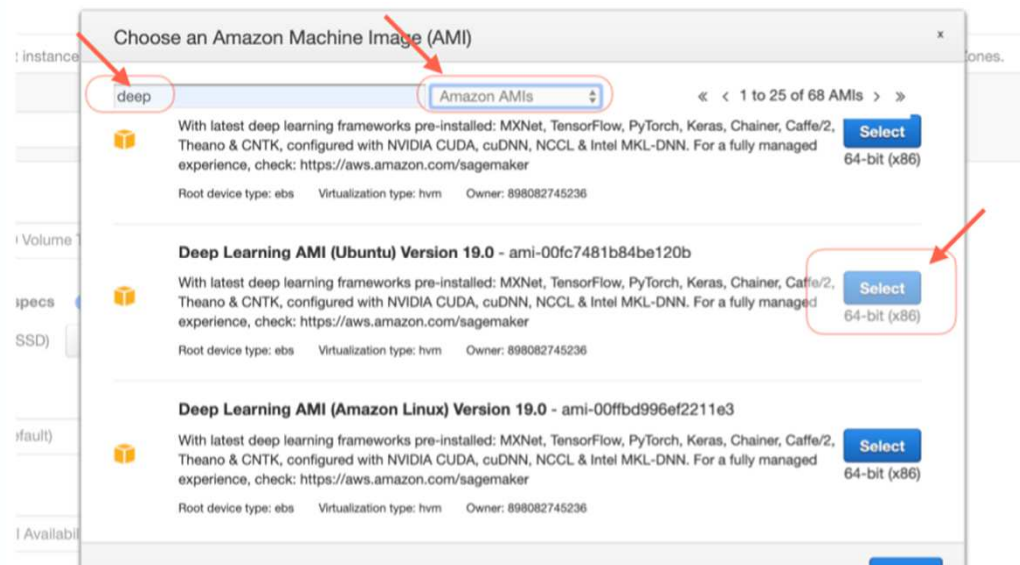
今回はコレにしてみる

AMIの選択

- ① 「カスタムAMIを使う」 ボタンを押す
- 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！

カスタム AMI を使う



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/>

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%

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	active	1 of 1	fulfilled	request	2 minutes ago	\$0.9
<input type="checkbox"/>	sir-rpjg5gmg	instance	p2.xlarge	active	i-0971be35980b...	fulfilled	one-time	2 minutes ago	\$0.9
<input type="checkbox"/>	sfr-d8051523-cdb1...	fleet	p2.xlarge	cancelled_...	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

The screenshot shows the AWS Management Console interface. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below these is a search bar and a table of instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4. Two instances are listed: one terminated and one running. A red circle with a '2' points to the radio button next to the running instance's name. Below the table, the details for the selected instance (i-0971be35980bc6585) are shown. The 'Description' tab is active, displaying various attributes. A red circle with a '3' points to the 'Security groups' attribute, which is set to 'default'. A link 'view inbound rules, view outbound rules' is visible next to 'default'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
	i-055ffc15dd9299f62	p2.xlarge	us-east-1e	terminated	2/2 checks ...	None	ec2-54-208-136-199.co...	54.20
	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			
Instance state	running			
Instance type	p2.xlarge			
Elastic IPs				
Availability zone	us-east-1e			
Security groups	default. view inbound rules, view outbound rules			
Scheduled events	No scheduled events			
AMI ID	Deep Learning AMI (Ubuntu) Version 19.0 (ami-00fc7481b84be120b)			
Public DNS (IPv4)	ec2-54-208-136-199.compute-1.amazonaws.com			
IPv4 Public IP	54.208.136.199			
IPv6 IPs	-			
Private DNS	ip-172-31-69-248.ec2.internal			
Private IPs	172.31.69.248			
Secondary private IPs				
VPC ID	vpc-48b8231			
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-48b78231 (default)

Security group rules:

Inbound Outbound

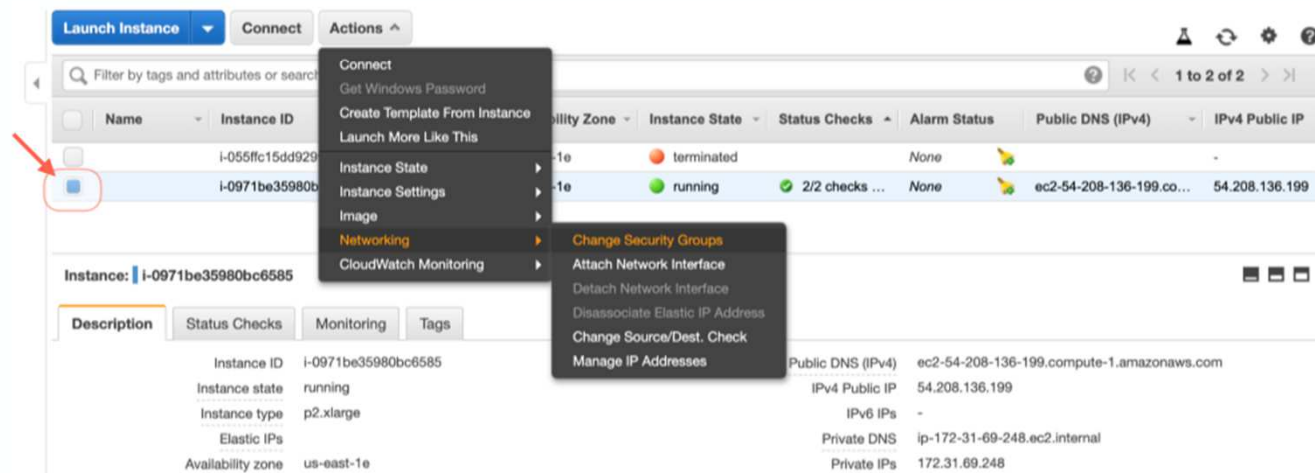
Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
Custom TCP ⓘ	TCP	8888	Anywhere ⓘ 0.0.0.0/0, ::0	e.g. SSH for Admin I
HTTP ⓘ	TCP	80	Anywhere ⓘ 0.0.0.0/0, ::0	e.g. SSH for Admin I
SSH ⓘ	TCP	22	Anywhere ⓘ 0.0.0.0/0, ::0	e.g. SSH for Admin I

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）



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

④yes/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 'ec2-54-208-136-139.compute-1.amazonaws.com (54.208.136.139)' can't be established.
ECDSA key fingerprint is SHA256:af636132c950b087c7375a0cM673ayHKA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-208-136-139.compute-1.amazonaws.com,54.208.136.139' (ECDSA) to the list of known hosts.

      _ _ _
     _ _ _ _ _
    _ _ _ _ _

Deep Learning MC (Ubuntu) Version 19.8
=====

Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1872-aws x86_64)

Please use one of the following commands to start the required environment with the frameworks of your choice:
for MXNet+Gluon+PyTorch (CUDA 9.0 and Intel MKL-DNN) ... source activate mxnet_3.08
for MXNet+Gluon+PyTorch (CUDA 9.0 and Intel MKL-DNN) ... source activate mxnet_3.08
for MXNet+Amazon Elastic Inference with PyTorch ... source activate amazon_elnet_3.08
for MXNet+Amazon Elastic Inference with PyTorch ... source activate amazon_elnet_3.08
for TensorFlow+Gluon+PyTorch (CUDA 9.0 and Intel MKL-DNN) ... source activate tensorflow_3.07
for TensorFlow+Gluon+PyTorch (CUDA 9.0 and Intel MKL-DNN) ... source activate tensorflow_3.07
for TensorFlow+Amazon Elastic Inference with PyTorch ... source activate amazon_tensorflow_3.07
for Theano+Gluon+PyTorch (CUDA 9.0) ... source activate theano_3.07
for Theano+Gluon+PyTorch (CUDA 9.0) ... source activate theano_3.07
for PyTorch with PyTorch (CUDA 9.0 and Intel MKL) ... source activate pytorch_3.08
for PyTorch with PyTorch (CUDA 9.0) ... source activate pytorch_3.08
for CNTK+Gluon+PyTorch (CUDA 9.0 and Intel MKL-DNN) ... source activate cntk_3.08
for CNTK+Gluon+PyTorch (CUDA 9.0 and Intel MKL-DNN) ... source activate cntk_3.08
for Caffe2 with PyTorch (CUDA 9.0) ... source activate caffe2_3.07
for Caffe2 with PyTorch (CUDA 9.0) ... source activate caffe2_3.07
for Caffe with PyTorch (CUDA 9.0) ... source activate caffe_3.08
for Chainer with PyTorch (CUDA 9.0 and Intel) ... source activate chainer_3.07
for Chainer with PyTorch (CUDA 9.0 and Intel) ... source activate chainer_3.07
for Base PyTorch (CUDA 9.0) ... source activate pytorch_3.08
for Base PyTorch (CUDA 9.0) ... source activate pytorch_3.08

Official Google User Guide: https://clouds.google.com/user-guides/index.html
AWS Deep Learning MC Homepage: https://aws.amazon.com/machine-learning/mc/
Developer Guide and Release Notes: https://docs.aws.amazon.com/dlami/latest/deepguide/what-is-dlami.html
Support: https://forums.aws.amazon.com/forum.jspa?forumID=246
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/support

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

90 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
aplicable law.
```

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

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...

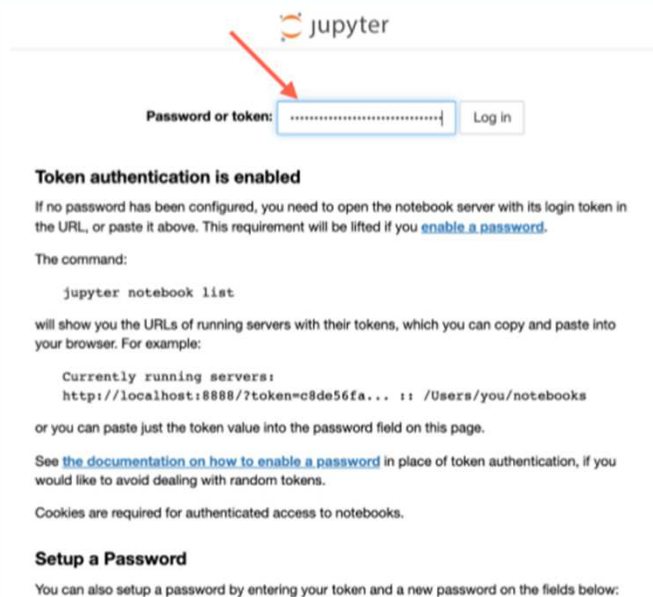
[23:32:50.619 NotebookApp] Found new kernels in environments: conda_cntk_p27, conda_amazonei_mxnet_p36, conda_caffe_p35, conda_theano_p36, conda. i. conda_caffe2_n27, conda_mxnet_n36, conda_mxnet_n27, conda_tensorflow_n36, conda_amazonei_mxnet_n27, conda_theano_n27, conda_nython2, conda_ana

token= 以下の文字列



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



The image shows the Jupyter Notebook login interface. At the top, there is a Jupyter logo and the word "jupyter". Below this, there is a label "Password or token:" followed by a text input field containing a series of dots. A red arrow points 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 open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if the user enables a password. The command "jupyter notebook list" is shown, followed by a paragraph stating that it will show the URLs of running servers with their tokens, which can be copied and pasted into the browser. An example URL is provided: "http://localhost:8888/?token=c8de56fa... : /Users/you/notebooks". It is noted that the user can paste just the token value into the password field. A link to the documentation on how to enable a password is provided. A note states that cookies are required for authenticated access to notebooks. Finally, there is a heading "Setup a Password" followed by a paragraph stating that the user can also setup a password by entering their token and a new password on the fields below.

jupyter

Password or token: Log in

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: