



# 大規模AIクラウド計算システム「ABCI」 初めてのABCI

2020年11月11日 国立研究開発法人 産業技術総合研究所 情報・人間工学領域 ディジタルアーキテクチャ推進センター

クイックレファレンス

https://abci.ai/ja/how to use/data/ABCI referrence.html



## 目次

#### <事前準備>

- 1. PC (Windows, Mac)へLinux, Python環境を構築 (Jupyter Notebook)
- WindowsPCの場合
   WSL (Windows Subsystem for Linux)によるLinux環境構築
- 3. 初回ログインとパスワード
- 4. 公開鍵の登録@利用ポータル
- 5. インタラクティブノードへのログイン (ABCI利用の入口)

#### <ハンズオン>

- 1. ABCI利用の概念的理解
- **2. 計算ノードにおける基本操作(Linux)** 
  - ジョブの投入 (qsub, qrsh)
  - **クラウドストレージの利用**
- 3. ABCI環境構築
  - ① Python仮想環境の構築
  - ② モジュールコマンド: ABCIにインストール済のソフトウェアを利用
    - 例: python, cuda, cudnn, singularity
  - ③ インストールコマンド:利用したいソフトウェアをインストール
    - 例: TensorFlow, Jupyter Notebook
  - ④ コンテナイメージの利用:ソフトウェア環境をそのまま利用
    - 例: digits
- 4. ABCI利用ポータル
- 5. ABCIサービス諸元





## PC (Windows, Mac)へLinux, Python環境を構築

Jupyter Notebookを利用する



### Anaconda3をインストール



- Anacondaは、PythonやJupyter Notebook等のパッケージを提供するプラットフォーム
- 下記のリンクからAnaconda3のインストーラをダウンロード

https://repo.continuum.io/archive/

\*頻繁にアップグレードされているので、新しいバージョンを選択

<MacOS> <Windows>

Anaconda3-2020.07-MacOSX-x86\_64.pkg Anaconda3-2020.07-Windows-x86.exe

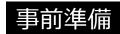
\*64bit版OSの場合は

Anaconda3-2020.07-Windows-x86 64.exe

- インストーラを実行
  - \*「インストール先」は、「自分専用(Just Me)」とすること。
  - \*インストール先のパスは「デフォルト」がお勧め(パスに日本語があるとうまく動作しない場合がある)。
  - \* Windowsでは、「Register Anaconda as my default Python 3.8」にチェックを入れるのがお勧め。



### Anaconda Navigatorを起動して Jupyter Notebookをインストール



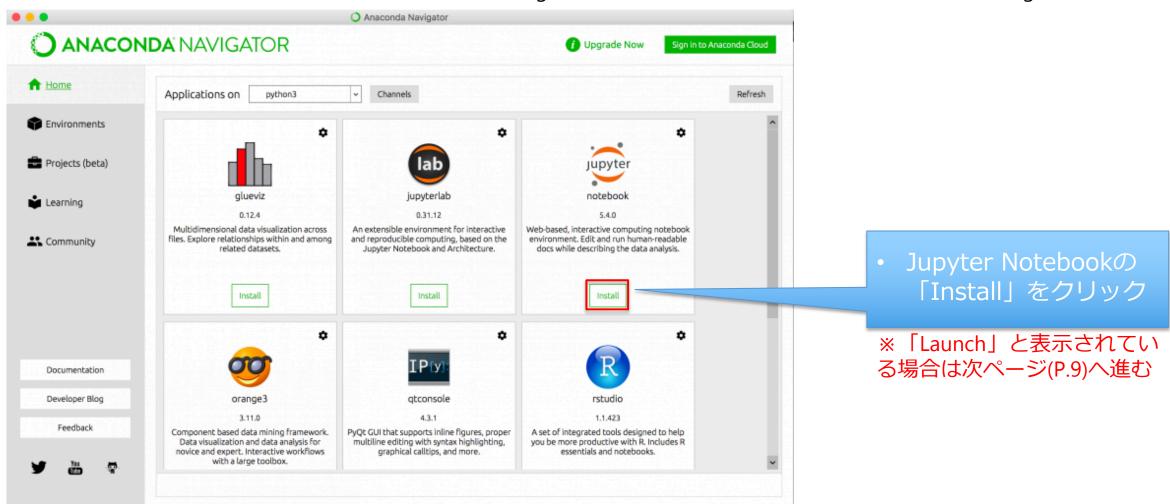
Anaconda Navigatorを起動

<MacOS>

<Windows>

Finder -> アプリケーション -> Anaconda-Navigator

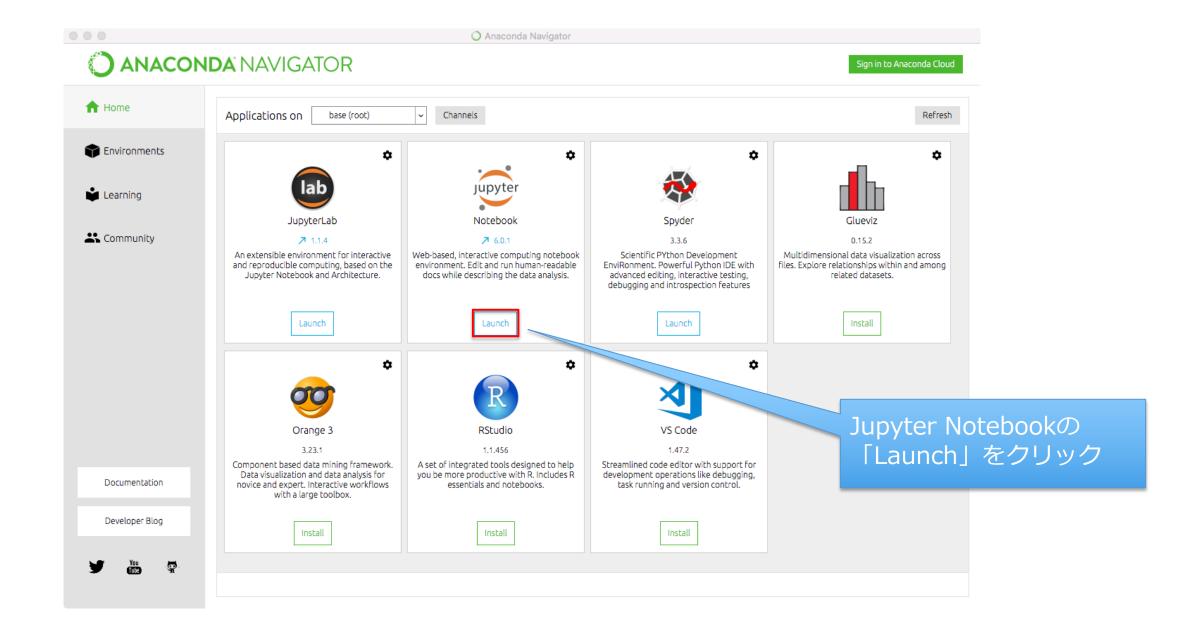
スタート -> Anaconda3 -> Anaconda Navigator

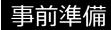




## Jupyter Notebookを起動









## Jupyter Notebook起動時にこのような画面が出たら

	💢 Jupyter
Password or token:	Log in
	Invalid credentials
Token authentication is enab	led
	ou need to open the notebook server with its login token in ment will be lifted if you enable a password.
The command:	
jupyter notebook list	
will show you the URLs of running serve your browser. For example:	ers with their tokens, which you can copy and paste into
Currently running servers: http://localhost:8888/?token=c	:8de56fa :: /Users/you/notebooks
or you can paste just the token value in	to the password field on this page.
See the documentation on how to en- would like to avoid dealing with random	able a password in place of token authentication, if you not tokens.
Cookies are required for authenticated	access to notebooks.
Setup a Password	
•	ring your token and a new password on the fields below:
Token	,
New Password	

Log in and set new password

(2) 一度パスワードを登録しておけば、 登録したパスワードを入力 (毎回、トークンをコピペする必要なし)

(1) ここに、トークンとパスワードを入力



### Jupyter Notebookのブラウザが自動で立ち上がらない場合

Jupyter Notebookを起動してもブラウザが自動で立ち上がらない場合は、 ターミナルに出力されたURLをブラウザにコピペする。

- インストールしたanaconda3のJupyter Notebookに再接続(「Logout」ボタンでログアウト 後に「login page」のリンクをクリックした場合等) した場合に、トークンを入力する画面 (P.10)が表示される場合がある。そのような使用を行う場合、事前に以下の手順でトークン を取得しておく。
  - 1. P.13の「Jupyter Notebookのターミナルを使う」の手順で、ターミナル画面を表示する。
  - 2. ターミナルで以下のコマンドを入力する。

#### jupyter notebook list

3. 以下のような結果が出力された場合、token=XX..XXの"XX..XX"の部分がトークンである。

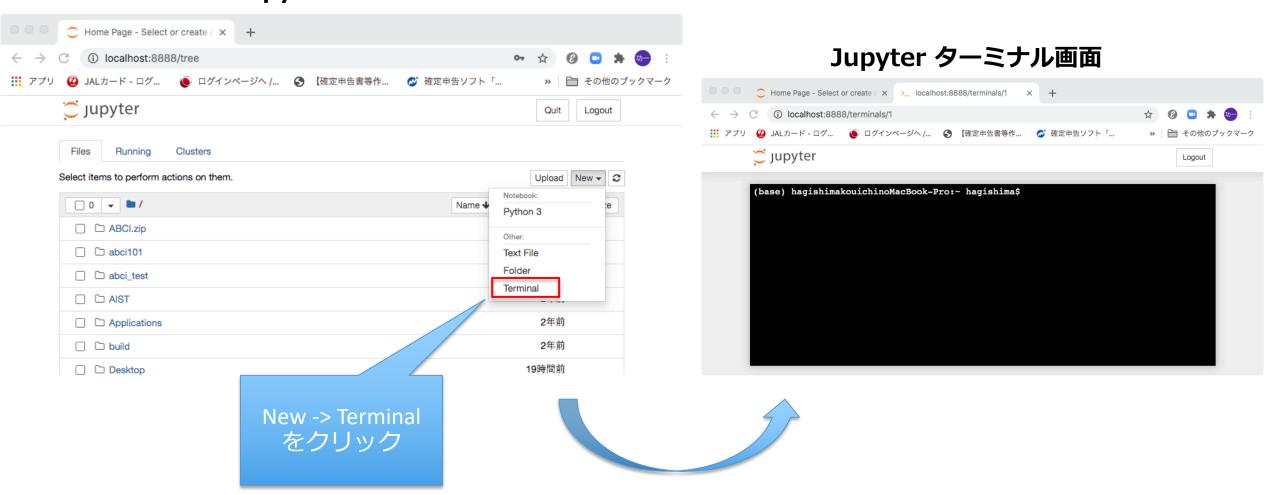
Currently running servers:
http://localhost:8888/?token=XX...XX ::



## **Jupyter Notebookのターミナルを使う**



#### Jupyter ホーム画面

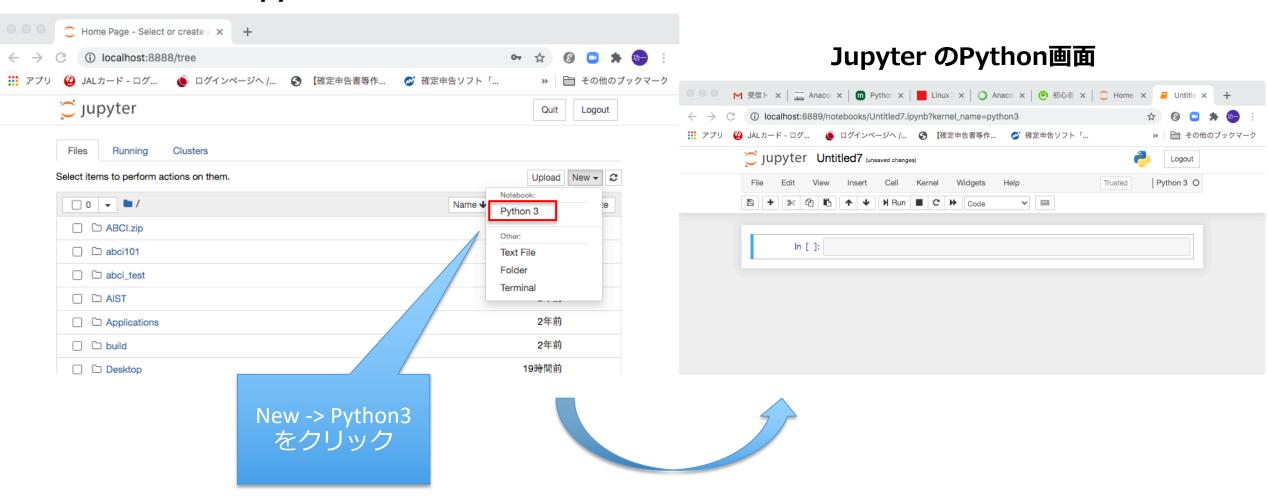




## Jupyter NotebookのPythonを使う



#### Jupyter ホーム画面





### Jupyter Notebookを終了させる

- Jupyter Notebookを起動した時に立ち上がったターミナルで、 「Control-C」を打つ。
- 以下のように聞かれるので、「y」を打つ。

The Jupyter Notebook is running at: http://localhost:8888/
Shutdown this notebook server (y/[n])?

#### (Windowsの場合)

• Jupyter Notebookの画面右上の「Quit」をクリックし、以下のメッセージが表示されたらブラウザのタブを閉じる

Server stopped ×

You have shut down Jupyter. You can now close this tab.

To use Jupyter again, you will need to relaunch it.



#### WSL (Windows Subsystem for Linux)によるLinux環境構築

- WSLを使えば、 WindowsPCでもターミナルでLinuxコマンドが使えるようになります。
- WSL (Windows Subsystem for Linux)によるLinux環境構築方法については、下記のサイト等をご参照ください。

https://qiita.com/tedkuma/items/01e5cfa09347176f4d1a



## 初回ログインURL、ABCIアカウント、仮パスワード をメール受信したら、、



## ABCI利用ポータルへの 初回ログインとパスワード登録

1. 初回ログインURLへアクセス パスワードは、「仮パスワード」を入力

利用者ログイン		
	ABCIアカウント名	
	パスワード	
	ログイン	

 利用規定への同意(右) 「同意する」をクリック後、 「全てを同意して次へ進む」をクリック

#### 利用規定への同意

下記を参照し、同意お願いします。同意いただけない場合ABCIをご利用できませんので、ご理解お願いいたします。

産総研外の利用者の方

産総研内の利用者の方

○同意する
●同意しない

以下のセキュリティ上の遵守項目を読み、同意いただけたら「全てを同意して次へ進む」で次に進んでください。

利用責任者、利用管理者、利用者(以後、利用者等という)は、研究所から提供されるABCI利用に関するアカウント及びアカウントのパスワードを研究所の承諾なく第三者に開示してはならず、かつ、第三者に推測されないように適切に設定し、管理しなければなりません。利用者等は、利用者等のデータ等がいかなる法令にも違反していないことを表明及び保証し、利用者等のデータ等の開発、内容、運用、維持及び利用につき、責任を負います。(利用者等のデータ等のセキュリティ及びバックアップ)

利用者等は、ABCIを適正に利用し、利用者等のデータ等について、セキュリティを確保し保護すること、及び定期的に保存することを含め、適切なセキュリティ及び保護を行うことを誓約します。(安全保障輸出管理関係法令の遵守)

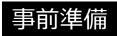
- 利用者等は、次の各号に該当する行為を行ってはなりません。
- 約款・規約及び回答書に記載されている事項に違反する行為 中請書に記載した利用目的以外にABCIを利用する行為
- 三 研究所若しくは第三者の著作権・商標権等の知的財産権を侵害する行為又はそのおそれがある行為
- 四 研究所若しくは第三者の財産、プライバシー若しくは肖像権を侵害する行為又はそのおそれがある行為
- 五 ABCIポイントを含めた研究所の電子情報を改ざん又は消去する行為
- 六 ウイルス等の有害なコンピュータプログラム等を開発する行為
- 土、研究所のネットワークやインターネット網、それらに接続されたサーバ設備等に不正にアクヤスする行為
- 八 ABCI利用サービスの提供を妨害する行為又は妨害するおそれのある行為
- 九 法令に違反する行為又はそのおそれがある行為
- 十 他の利用者や第三者に著しく迷惑をかけ、又は社会的に許されないような行為

3. <i>/</i>	パスワー	ド変史

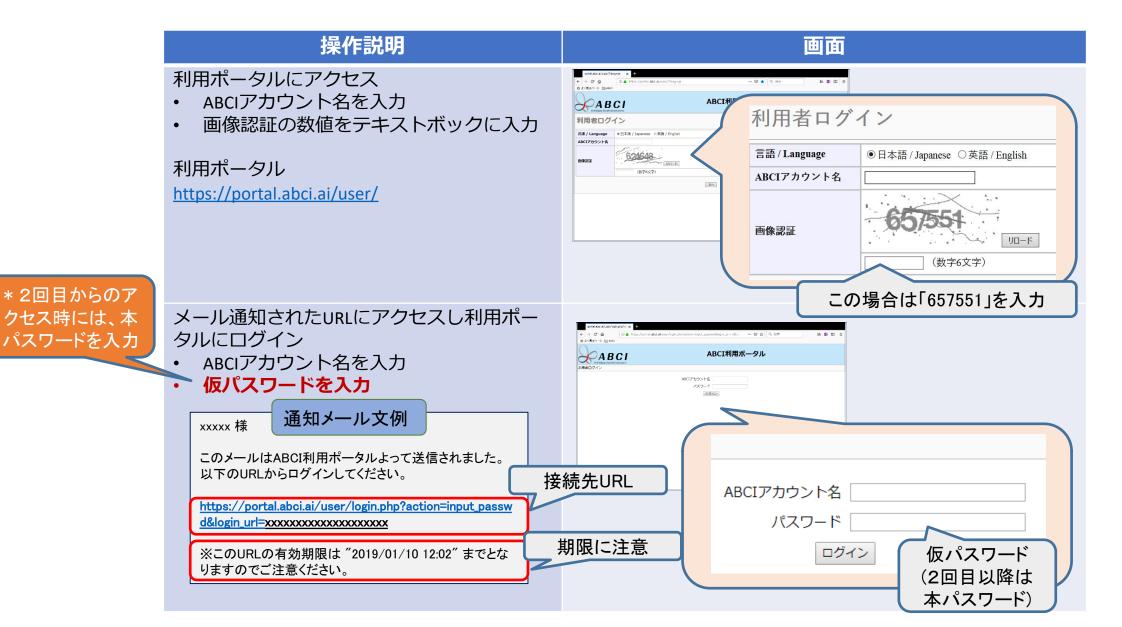
パスワード変更				
以下のフォームにパスワードをご入力の上、「変更」ボタンをクリックしてください。				
現在のパスワード				
新しいパスワード				
新しいパスワード (確認)				
(アスワード規約)  15以上の文字をランダムに並べた文字列を指定してください。例えばLinuxの辞書に登録されている単語は使用できません。文字をランダムに選ぶ方法として、パスワード作成用のソフトウェアを用いるなどして、自動的に生成することを推奨します。  現在のパスワードと異なる文字列を指定してください。  英小文字、英大文字、数字、特殊文字の4種類全てを含む文字列を指定してください。  使用可能な特殊文字は以下の通りです。 空白、!、"、#、\$、%、&、'、(、)、*、+、,、-、、/、:、;、<、=、>、?、@、[、¥、]、^、_、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、				

全てを同意して次へ進む





## 初期アクセスURLがタイムアウトになってしまったら、「







## ABCI利用開始前に公開鍵を登録する(必須)





### 公開鍵・秘密鍵ペアの生成

1) ターミナルを開き、公開鍵・秘密鍵のペアを生成する(ssh-rsaの例)。 「<u>xxx@yyy.co.jp</u>」/は、自分のメルアドと置き換える!

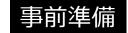
```
yourpc$ install -m 0700 -d ~/.ssh
yourpc$ cd ~/.ssh
yourpc$ ssh-keygen -t rsa -b 4096 -C "xxx@yyy.co.jp" -f ~/.ssh/id_rsa
Generating public/private rsa key pair. Enter passphrase (empty for no passphrase): <-- パスフレーズを入力(任意)
Enter same passphrase again: <-- もう一度、パスフレーズを入力(任意)
Your identification has been saved in id_rsa.
Your public key has been saved in id_rsa.pub.
yourpc$ Is ~/.ssh
id_rsa id_rsa.pub
```

- \* id\_rsa (秘密鍵)・・・ クライアントPCに配置
- \* id\_rsa.pub (公開鍵)・・・ ABCIに登録
- 2) ターミナルから公開鍵を参照する(ssh-rsaの例)。

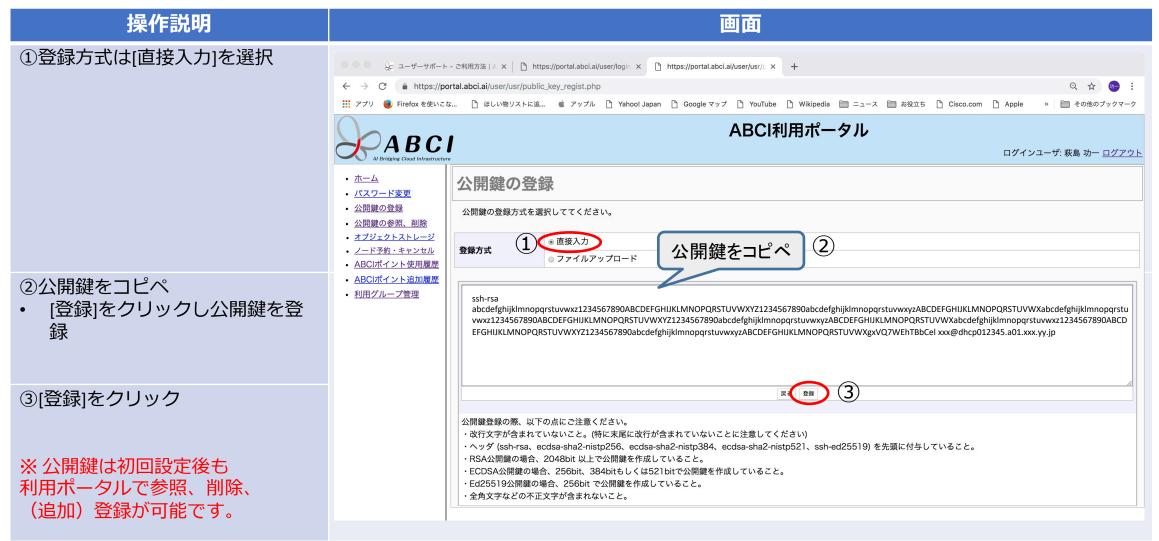
\* 公開鍵は架空のものです。

```
yourpc$ cat ~/.ssh/id_rsa.pub ssh-rsa abcdefghijklmnopqrstuvwxz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXgxVQ7WEhTBbCel xxx@dhcp012345.a01.yyy.co.jp
```



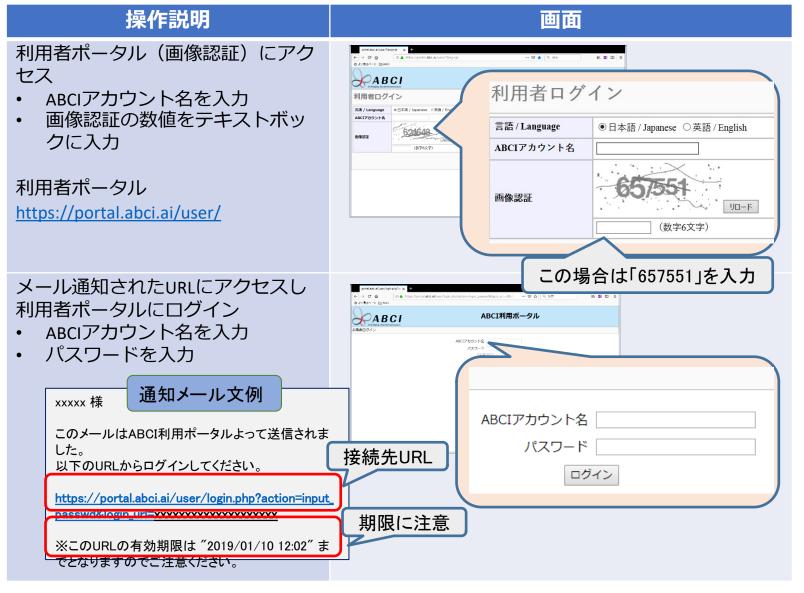


### 公開鍵の登録 ABCI利用ポータル





### ABCI利用ポータル



#### ABCI利用者の設定・参照:

- パスワード変更
- 公開鍵の登録
- 公開鍵の参照、削除
- ノード予約・キャンセル(利用者に係るもの)
- ABCIポイント使用履歴
- ABCIポイント追加履歴

https://docs.abci.ai/portal/ja/02/

#### ABCI利用グループの管理:

(利用責任者又は利用管理者権限が必要)

- ABCIポイントの追加申請
- グループディスクの追加申請
- 利用者の追加申請
- 利用者の編集
- 利用者の削除
- 利用管理者権限の設定・削除
- 利用責任者の変更申請

https://docs.abci.ai/portal/ja/03

#### クラウドストレージ管理 (New):

• アカウント管理





## ABCI利用ポータルヘアクセスし、 自分が所属する利用グループを確認する





### インタラクティブノードへのログイン

Windowsユーザ(かつWSLを導入されていない)の皆様は、インタラクティブノードへのログインは、SSHクライアントソフトウェア (PuTTY, Teraterm等)を使用願います。 具体的な接続方法については、ABCI User Guideを参照願います。

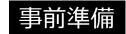
Teratermの利用: https://docs.abci.ai/ja/tips/tera-term/

PuTTYの利用: https://docs.abci.ai/ja/tips/putty/

\*所属する機関によっては下記のようなセキュリティポリシーを適用している場合があります。その際は、所属する機関のIT部門からのサポートを得てください。

- SSH接続をプロキシー経由としている
- 秘密鍵を(各自のPCではなく)サーバーに置くことになっている
- アクセス制御により、下記のようなポート番号を使用禁止にしている。
  - 10022, 8888, 6006 ... (ABCIで利用するポート番号)





# インタラクティブノードへのログイン(ssh)

「aaa12345xx」は、自分の「ABCI-ID」に置き換える!

1) ターミナルからアクセスサーバ(as.abci.ai)にログインします。

yourpc\$ ssh -L 10022:es.22 -l aaa12345xx as.abci.ai

The authenticity of host 'as.abci.ai (0.0.0.1)' can't be established.

Are you sure you want to continue connecting (yes/no)? ← yesを入力

Warning: Permanently added 'XX.XX.XX.XX' (RSA) to the list of known hosts.

Enter passphrase for key '/home/username/.ssh/id\_rsa': ← パスフレーズ入力

Welcome to ABCI access server. Please press any key if you disconnect this session.

#### **Warning**

\* 秘密鍵ファイルへのパス指定サブコマンド: -i~/.ssh/秘密鍵ファイル名

上記状態で何らかのキーを入力するとSSH接続が切断されてしますので注意してください。

2) 別のターミナルで、インタラクティブノード(es)にポートフォーワーディングします。

#### yourpc\$ ssh -p 10022 -l aaa12345xx localhost

Are you sure you want to continue connecting (yes/no)? yes ← yesを入力

Warning: Permanently added 'localhost' (RSA) to the list of known hosts.

Enter passphrase for key '/home/username/.ssh/id\_rsa': ← パスフレーズ入力

[username@es1 ~]\$

#### **Warning**

終了処理:終了する場合は、「exit」コマンドを投入。何も操作がなければ、1時間でセッションは自動的に切断されます。

ssh -L *10022:es:22* -l aaa12345xx as.abci.ai ssh -p *10022* -l aaa12345xx localhost

localhostのポート番号:remote host:remote hostのポート番号 localhostのポート番号(ssh -Lで使った番号と一致させる)



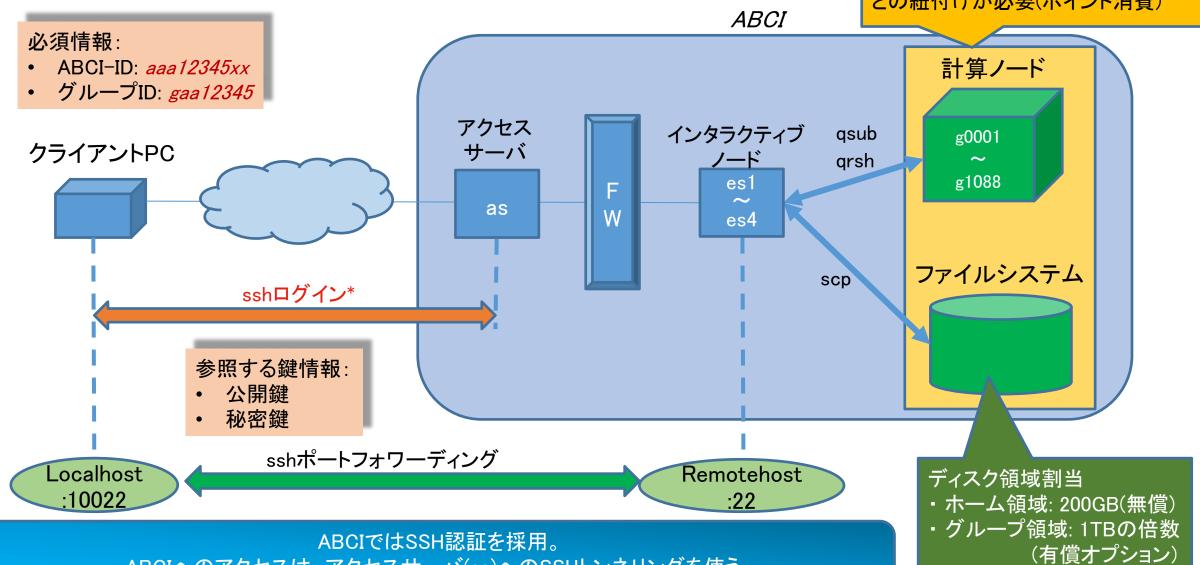
### \*ハンズオンはここからスタートします。

これで、ABCIを利用する準備が整いました



### ABCI利用概念図

これらのリソース使用時には、 必ず、グループID との紐付けが必要(ポイント消費)

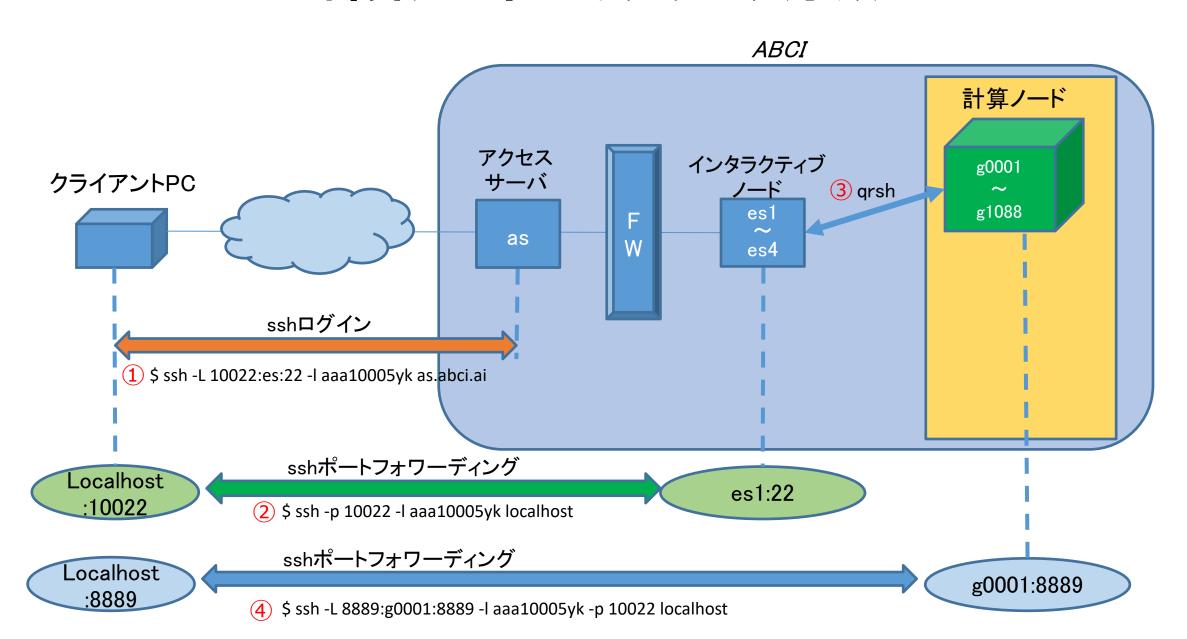


ABCIへのアクセスは、アクセスサーバ(as)へのSSHトンネリングを使う。 \*このSSHセッションはABCI利用終了まで接続を切ってはいけない。

24



### 計算ノードへのアクセス手順



### ABCIでのオペレーション

- ABCIはLinux OSで構成されたシステム
  - コマンド実行による対話的な操作が主体



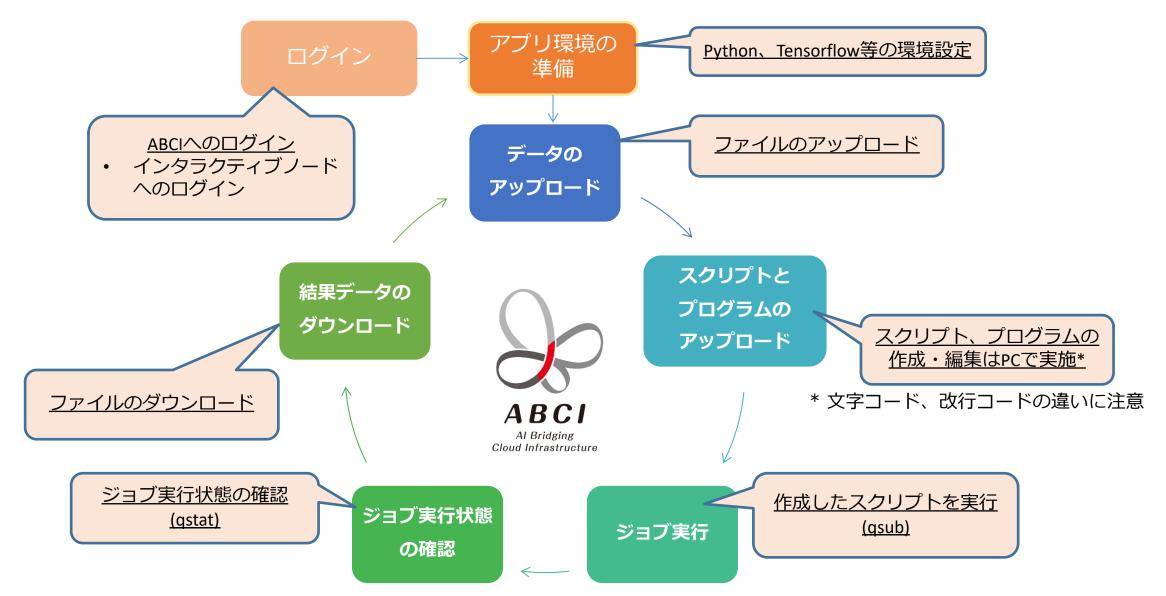
定型の操作を覚えれば利用は簡単



## 計算ノードにおける基本操作

- ジョブの投入 (qsub, qrsh)
- ファイルのアップロード、ダウンロード (scp)
- クラウドストレージの利用

## バッチジョブによるABCI利用フロー





### バッチジョブの準備 ジョブスクリプトの作成と実行権の付与



1) PCでジョブスクリプト(Sample.sh)を作成し、PCのホーム領域に保存。

date
hostname
echo "Hello, world"

2) PC環境で作成したジョブスクリプト(Sample.sh)をABCIへアップロード。「aaa12345xx」は、自分の「ABCI-ID」に置き換える!

```
yourpc$ scp -P 10022 Sample.sh aaa12345xx@localhost:Sample.sh
Enter passphrase for key: ++++++++ ← パスフレーズを入力
Sample.sh 100% file-size transfer-speed transfer-time
```

3-1) ジョブスクリプト(Sample.sh)の改行コードがCR+LFまたはCRになっている場合: 改行コードをLinux環境(LF)に変更し、sample.shで保存。

```
[username@es1 ~]$ nkf -Lu Sample.sh > sample.sh
```

3-2) 文字コードがShift-JIS、改行コードがCR+LFになっている場合: 文字コードと改行コードをLinux環境(UTF-8, LF)に一括変換。

```
[username@es1 ~]$ nkf -wd Sample.sh > sample.sh
```

3-3) 文字コード、改行コードの確認。

```
[username@es1 ~]$ nkf --guess sample.sh sample.sh: UTF-8 (LF)
```

4) インタラクティブノード(es)にログインして、ジョブスクリプトに実行権を付与。

```
[username@es1~]$ chmod u+x sample.sh ジョブスクリプトに実行権を付与
[username@es1~]$ ls -l

total 0

rwxr--r-- 1 username usergroup 61 Jan 11 13:12 sample.sh
```

- \* Linuxでは、文字コード(UTF-8)、改行コード(LF)が使用される。PCのOSによっては文字コード、改行コードの変換が必要。
- ・ Windowsの場合: 文字コード(Shift-JIS)、改行コード(CR+LF)
- Mac OS X以前のMacの場合: 改行コード(CR)





## バッチジョブ(qsub)

1) インタラクティブノード(es)にログインして、sample.shを実行。

「gaa12345xx」は、自分の「利用グループ名」に置き換える!

[aaa12345xx@es1~]\$ qsub -g gaa12345 -l rt\_C.small=1 sample.sh # gaa12345: グループ名, rt C.small=1: 計算資源タイプ(CPU x 5コア), sample.sh: ジョブスクリプト Your job 151645 ("sample.sh") has been submitted

2) バッチジョブの状況を参照。

```
[aaa12345xx@es1~]$ qstat
iob-ID
        prior
                                     state submit/start at
                                                                                  jclass slots ja-task-ID
                name
                           user
                                                                queue
151646 0.25586 sample.sh aaa12345xx r 01/20/2019 15:16:53
                                                                                         10
                                                               gpu@g0002
```

3) バッチジョブの出力(ホームディレクトリ)。

```
[aaa12345xx@es1 ~]$ Is -I
-rw-r--r-- 1 aaa12345xx gaa12345
                                 172 1月 20 15:17 sample.sh.e151646 #エラー出力ファイル(数字はジョブ番号)
                                   0 1月 20 13:51 sample.sh.o151235 #正常出力ファイル(数字はジョブ番号)
-rw-r--r-- 1 aaa12345xx gaa12345
```

予約ノードの指定法: サブコマンド( -ar *ar\_id* )

# ar\_id は「予約ID」(4桁の数字)

#予約IDは、「利用ポータル」の「ノード予約・キャンセル」または、grstatコマンドで参照できる。

優先度を上げるには: サブコマンド(-p-400)

# ポイントは通常の1.5倍

ジョブをプログラム(スクリプト)化 -> 多数のジョブを一度に投げられる





# インタラクティブジョブ(qrsh)

1) インタラクティブノード(es)にログインして実行。

「gaa12345xx」は、自分の「利用グループ名」に置き換える!

[aaa12345xx@es1~]\$ qrsh -g gaa12345 -l rt\_F=1 -l h\_rt=01:00:00 # gaa12345: グループ名, rt\_F=1 : 計算資源タイプ(フルノードを1個), h\_rt=01:00:00(最大1時間確保)

2) インタラクティブジョブの状況を参照。

[aaa1234 job−ID	l5xx@es1 î prior	]\$ qstat name	user	state	submit/start at	queue	jclass	slots ja-task-ID
151646	0.28027	QRLOGIN	aaa12345xx	r	01/21/2019 09:39:43	gpu@g0371		10

予約ノードの指定法: サブコマンド (-ar *ar\_id*)

# ar id は「予約ID」(4桁の数字)

#予約IDは、「利用ポータル」の「ノード予約・キャンセル」または、grstatコマンドで参照できる。

ジョブ実行時に、インタラクティブに計算ノードへアクセス可能 -> デバッグ等に便利



# ファイルのアップロード・ダウンロード(scp)



1) ファイルのアップロード: インタラクティブノード(es)にログインし、別のターミナルからscpを実行。

```
yourpc$ scp −P 10022 local−file aaa12345xx@localhost:remote−file # local−file : PCからアップしたいファイル、remote−file : リモートファイル名 Enter passphrase for key: +++++++ ← パスフレーズを入力 local−file 100% file−size transfer−speed transfer−time
```

「aaa12345xx」は、自分の「ABCI-ID」に置き換える!

2) ファイルのダウンロード: インタラクティブノード(es)にログインし、別のターミナルからscpを実行。

```
yourpc$ scp -P 10022 aaa12345xx@localhost:sample ./
# sample: PCへダウンロードしたいファイル
Enter passphrase for key: +++++++ ← パスフレーズを入力
sample 100% file-size transfer-speed transfer-time
```



# ABCIクラウドストレージの特徴

#### Amazon S3 互換のオブジェクトストレージサービス

- ① AWS CLI (コマンドラインインタフェース) が使える
- ② ABCIにログインしなくても、httpsで外部から使える
- ③ サーバーサイド暗号化に対応している
- ④ 複数のクラウドストレージ・アカウントを持てる
- ⑤ バケット単位で、アクセス制御を設定できる (非公開、ABCI利用グループで共有、全世界への公開を選択できる)



### ABCIクラウドストレージの利用例

#### ■ <ユースケース1>データやプログラムの共有・公開の場として

データやプログラムを特定のグループ内(又は複数のグループ間)で共有。バケット単位で自由にアクセス制御ができるので、 柔軟に共有ができる。特に、クラウドストレージに保存されたデータを利用してABCIの計算ノードで処理する場合、データ転送を 低レイテンシーでゼロコストで可能となる。

#### ■ <ユースケース2>IoTデバイスやエッジシステムとのデータ連携として

カメラやセンサーなどのIoTデバイスで取得したデータを直接、ABCIクラウドストレージに格納。取得したばかりのデータを使って、ABCI計算ノードで処理させることができる。

#### ■ <ユースケース3>ホーム領域の延長として

ホーム領域が不足してきた場合、データの保管場所として利用。低額の従量制課金なので、グループ領域(共有ディスク)をTB 単位、月単位で使用するよりも、低料金で柔軟に使うことができる。

#### ■ <ユースケース4>複数のアカウントで使い分ける

研究者は複数のプロジェクトを推進している場合が殆ど。複数のアカウントを作れるので、プロジェクト毎にバケットの管理 (アクセス制御)を行うことができる。

#### ■ <ユースケース5>暗号化してデータの情報漏洩を防止する

サーバーサイドの暗号化機能を使えば、万一、クラウドストレージから情報が漏洩してもデータを復元できない。



### ABCIクラウドストレージの使い方



https://docs.abci.ai/ja/abci-cloudstorage/usage/ https://docs.abci.ai/portal/ja/03/#manage-cloudstorage

• AWS-CLIモジュールのロード

[username@es1 ~]\$ module load aws-cli

#### • 各 種 操 作

• 認証情報などの設定

[username@es1 ~]\$ aws configure AWS Access Key ID [None]: ACCESS-KEY-ID AWS Secret Access Key [None]: SECRET-ACCESS-KEY Default region name [None]: us-east-1 Default output format [None]:(入力不要)

```
AWS-CLI構文aws [options] < command > < subcommand > [parameters]
  例
                 aws --endpoint-url https://s3.abci.ai s3 ls
                 s3: コマンド、ls: サブコマンド
バケットの作成:
      [username@es1 ~]$ aws --endpoint-url https://s3.abci.ai s3 mb
s3://dataset-summer-2012
      make bucket: dataset-summer-2012
バケットの一覧:
      [username@es1 ~]$ aws --endpoint-url https://s3.abci.ai s3 ls
      2019-06-15 10:47:37 testbucket1
      2019-06-15 18:10:37 testbucket2
オブジェクトのリスト:
      [username@es1 ~]$ aws --endpoint-url https://s3.abci.ai s3 ls
s3://testbucket1
      2019-07-05 17:33:05 4 test1.txt
```

2019-07-05 21:12:47 4 test2.txt

```
データのコピー:
      [username@es1 ~]$ aws --endpoint-url https://s3.abci.ai s3 cp ./images/0001.jpg
s3://dataset-c0541/
      upload: images/0001.jpg to s3://dataset-c0541/0001.jpg
データの移動:
      [username@es1 ~]$ aws --endpoint-url https://s3.abci.ai s3 mv annotations.zip
s3://dataset-c0541/
      move: ./annotations.zip to s3://dataset-c0541/annotations.zip
オブジェクトのリスト:
      [username@es1 ~]$ aws --endpoint-url https://s3.abci.ai s3 ls s3://testbucket1
      2019-07-05 17:33:05 4 test1.txt
      2019-07-05 21:12:47 4 test2.txt
ローカルディレクトリとクラウドストレージの同期:
      [username@es1 ~]$ aws --endpoint-url https://s3.abci.ai s3 sync ./sensor2
s3://mvbucket/
      upload: sensor2/0002.dat to s3://mybucket/0002.dat
      upload: sensor2/0004.dat to s3://mybucket/0004.dat
オブジェクトの削除:
      [username@es1 ~]$ aws --endpoint-url https://s3.abci.ai s3 rm
s3://mvbucket/readme.txt
      delete: s3://mybucket/readme.txt
```



## さあ、ABCIを利用してみよう! ABCI環境構築

- モジュールコマンド: ABCIにインストール済のソフトウェアを利用例: python, cuda, cudnn, singularity
- インストールコマンド:利用したいソフトウェアをインストール 例: TensorFlow, Jupyter Notebook
- コンテナイメージの利用:ソフトウェア利用環境をそのまま利用 例:digits



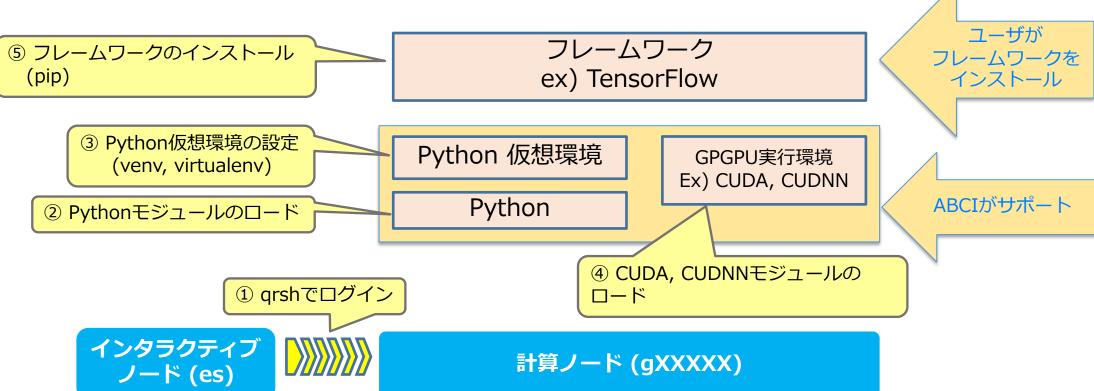


インストールを1回だけ実施すれば、

ホーム領域に保存される

### アプリケーション環境の構築

- ① インタラクティブジョブ(qrsh)で計算ノードへログイン
- ② Pythonモジュールのロード
- ③ Python仮想環境の設定
- ④ GPGPU実行環境の設定
- ⑤ アプリケーション環境(ex. フレームワーク)の設定

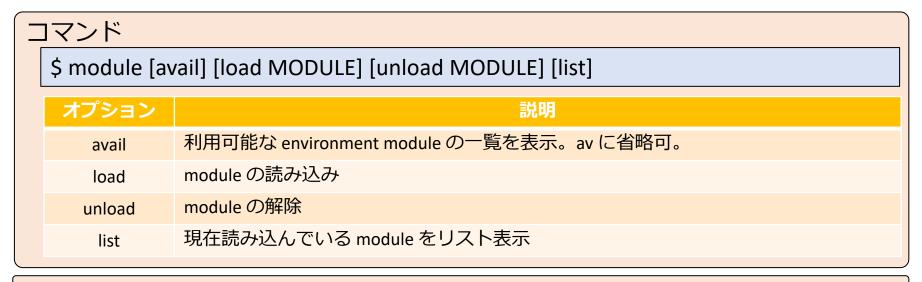


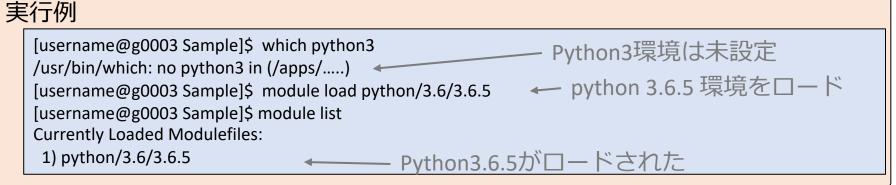


### Pythonモジュールのロード



- アプリ実行環境の適用
  - ABCIで用意されたアプリ実行環境(環境変数)を設定可能







### Python仮想環境の設定

- プロジェクトのアプリ実行環境をディレクトリ単位で管理可能 (他アプリ実行環境とのバージョン干渉を排除できる)
- python2 系は virtualenv, python3 系は venv など多数あり
- 仮想環境を使う度に activate で有効化、deactivate で終了

```
実行例: virtualenv (python2)
[username@g0003 ~]$ virtualenv ~/Sample/v tf gpu ← 仮想環境を作成。ディレクトリapp1を作成
[username@q0003 ~]$ source ~/Sample/v_tf_gpu/bin/activate
                                                       仮想環境を開始。(csh の場合 activate.csh)
(v tf gpu)[username@g0003 ~]$ pip install ...
                                               この間、pip などで取得するモジュールの
(v tf gpu)[username@g0003 \sim]$:
                                              ファイルは ~/Sample/v_tf_gpuディレクトリに配置される。
(v tf gpu)[username@g0003 ~]$ deactivate ←
                                                    仮想環境を終了。
```

#### 実行例: venv (python3)

```
仮想環境を作成。ディレクトリapp2を作成
[username@g0003 ~]$ python3 -m venv ~/Sample/v tf gpu-
[username@g0003 ~]$ source ~/Sample/v tf gpu/bin/activate
                                                        仮想環境を開始。(csh の場合 activate.csh)
(v tf gpu)[[username@g0003 ~]$ pip install ...
                                                この間、pip などで取得するモジュールの
(v tf gpu)[[username@g0003 \sim]$
                                                ファイルは ~/Sample/v_tf_gpuディレクトリに配置される。
(v tf gpu)[[username@g0003 ~]$ deactivate ___
                                                        仮想環境を終了。
```



### ABCIでJupyter Notebookを使う

https://docs.abci.ai/ja/tips/jupyter-notebook/



### ABCIでJupyter Notebookを起動

(1) 計算ノードを一台占有し、Python仮想環境を作成し、 pipでtensorflow-gpuとjupyterをインストール(一度、実施すればいい)。

```
[username@es3 ~]$ grsh -g gaa12345 -l rt F=1
[username@g0001 ~]$ module load python/3.6/3.6.5 cuda/10.0/10.0.130.1 cudnn/7.4/7.4.2
[username@g0001 ~]$ python3 -m venv ~/jupyter_env
[username@g0001 ~]$ source ~/jupyter_env/bin/activate
(jupyter_env)[username@g0001 ~]$ pip3 install tensorflow-gpu jupyter numpy==1.16.4
```

\* 次回以降は、以下のようにモジュールの読み込みと~/jupyter\_envのアクティベートだけでいい。

```
[username@es3 ~]$ qrsh -g gaa12345 -l rt_F=1
[username@g0001 \sim]$ module load python/3.6/3.6.5 cuda/10.0/10.0.130.1 cudnn/7.4/7.4.2
[username@q0001 ~]$ source ~/jupyter env/bin/activate
(jupyter_env)[username@g0001 ~]
```

(2) Jupyter Notebookを起動。

or http://127.0.0.1:8888/?token=token string

ローカルPCでJupyterを起動中でポート8888を使用している場合、「8889」を使用!

```
(jupyter_env)[username@g0001 ~]$ jupyter notebook --ip=`hostname` --port=8888 --no-browser
[I 20:41:12.082 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 20:41:12.090 NotebookApp]
To access the notebook, open this file in a browser:
file:///home/username/.local/share/jupyter/runtime/nbserver-xxxxxx-open.html
Or copy and paste one of these URLs:
http://g0001.abci.local:8888/?token=token_string
```

### Jupyter NotebookでABCIを利用

(3) 別ターミナルで。ローカルPCの8888番ポートを計算ノードの8888番ポートに 転送するSSHトンネルを作成。

yourpc\$ ssh -L 8888:g0001:8888 -l username -p 10022 localhost

(4) ブラウザで下記のURLを開く(トークンは、(2)をコピペ)。

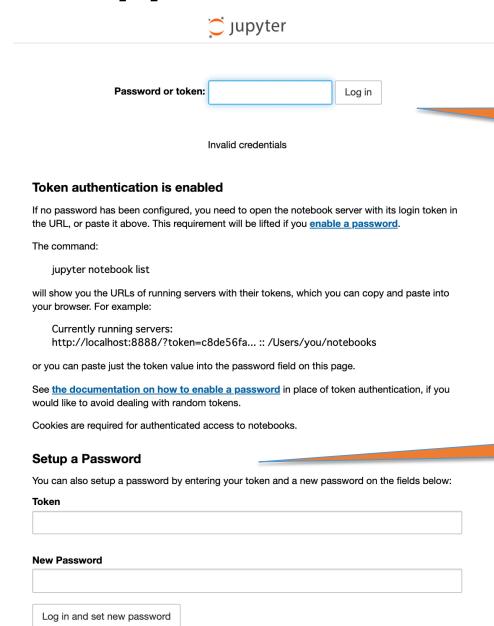
http://127.0.0.1:8888/?token=token\_string

- \* パスワードを登録しておけば、毎回トークンをコピペする必要はない。 (詳細は、次ページを参照)
- \* Jupyter Notebook利用上の注意点 ## notebookコンフィグファイル Jupyter Notebookの利用には、notebookコンフィグファイルが必要です。 「jupyter\_notebook\_config.py」が下記のフォルダにない場合は、次のコマンドで notebookコンフィグファイルを作成してください。
- Windows: C:\(\text{Users\(\text{\text{USERNAME\(\text{\text{\text{\text{\text{upyter\(\text{\tin}\text{\tex
- OS X: /Users/USERNAME/.jupyter/jupyter\_notebook\_config.py
- Linux: /home/USERNAME/.jupyter/jupyter\_notebook\_config.py

yourpc\$ jupyter notebook --generate-config



### **Jupyter Notebook: このような画面が出たら**

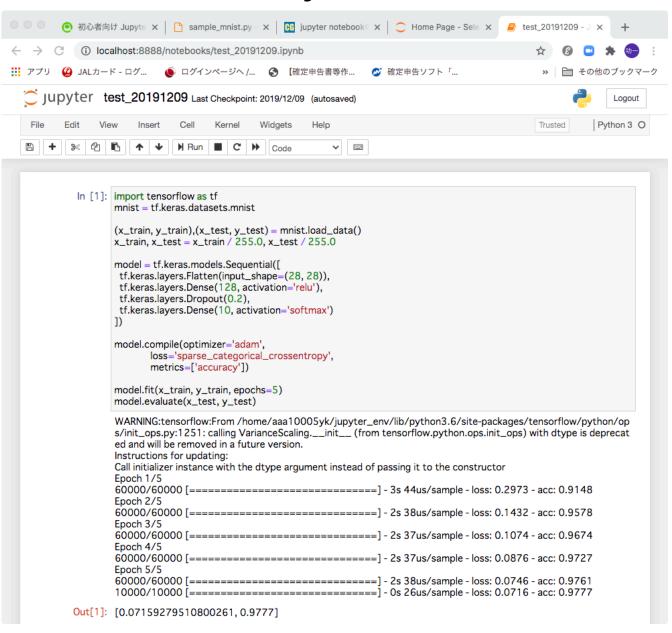


(2) 一度パスワードを登録しておけば、 登録したパスワードを入力 (毎回、トークンをコピペする必要なし)

(1) ここに、トークンとパスワードを入力



### Jupyter NotebookでPythonプログラミング例



45 **AIST** 

### Jupyter Notebookの終了

#### 以下の手順で終了。

- ローカルPCのJupyter Notebookホーム画面の「quit」ボタンで終了。
- ローカルPCの8888ポートを転送していたSSHトンネル接続を Control-C で終了。
- 計算ノードで起動中のJupyterプログラムが終了してない場合は、 Control-C で終了。



### コンテナ(Singularity)の利用

https://docs.abci.ai/ja/09/



### ABCIで利用可能なコンテナは、Singularity

- Singularity
  - ABCIではLinuxコンテナとしてSingularityが利用可能
  - 利用可能なバージョンは、moduleコマンドで確認可
  - NGC(※)コンテナレジストリなどから、Docker コンテナイメージ をSingularityイメージに変換し利用が可能 GPUに最適化されたディープラーニングフレームワークコンテナやHPCア プリケーションコンテナのDockerイメージと、それらを配布するための NGCコンテナレジストリを提供

```
実行例(module)
$ module avail
         ----- /apps/modules/modulefiles/runtimes ------
               singularity/2.6.1 spark/2.3.2
hadoop/2.9.1
hadoop/2.9.2
               spark/2.3.1
                           spark/2.4.0
```

### Singularityイメージの作成

NVIDIA NGCからDIGITSを取得

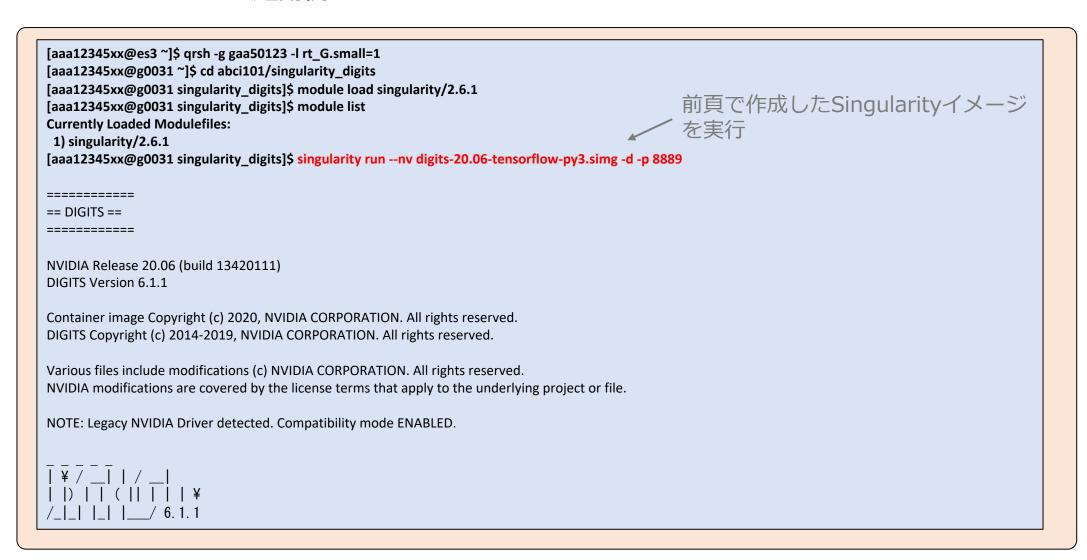
#### NGC コンテナレジストリからDIGITSイメージを取得し、Singularityイメージを作成

```
[aaa12345xx@es3 ~]$ cd ~/abci101
[aaa12345xx@es3 abci101]$ mkdir singularity_digits
[aaa12345xx@es3 abci101]$ cd singularity digits
[aaa12345xx@es3 singularity_digits]$ module load singularity/2.6.1
[aaa12345xx@es3 singularity_digits]$ singularity pull docker://nvcr.io/nvidia/digits:20.06-
tensorflow-pv3
                                                  イメージファイルの取得と作成
Docker image path: nvcr.io/nvidia/digits:20.06-tensorflow-py3
Setting 700 permissions on folder /home/aaa12345xx/.singularity/docker
Cache folder set to /home/aaa12345xx/.singularity/docker
くく中略>>
Building Singularity image...
Singularity container built: ./digits-20.06-tensorflow-py3.simg
                                                     Docker イメージが変換され
Cleaning up...
Done. Container is at: ./digits-20.06-tensorflow-py3.simg / .simg ファイルが作成される
[aaa12345xx@es3 singularity digits]$
```



### Singularityイメージの起動

DIGITSの起動例





### ABCI利用ポータル

https://portal.abci.ai/user/





portal.abci.ai/user/login.php?action=input\_passwd&login\_url=6b6a0ff11b4aa03e639c72a12712e2657bdab6cae114172d989c48b5d6...



### ABCI利用ポータル

利用者ログイン

ABCIアカウント名

パスワード

ログイン

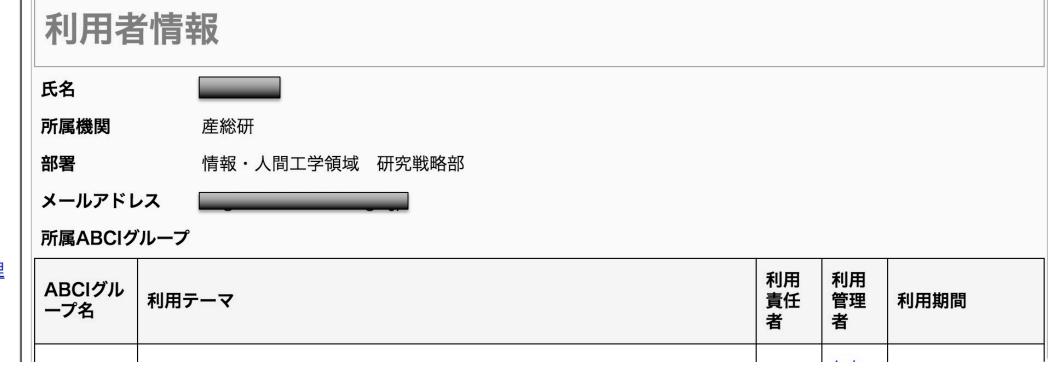






### ABCI利用ポータル

- ホーム
- パスワード変更
- 公開鍵の登録
- 公開鍵の参照、削除
- ノード予約・キャンセル
- ABCIポイント使用履歴
- ABCIポイント追加履歴
- 利用グループ管理
- クラウドストレージ管理









## ABCIサービス諸元



New! =

### 1. 計算ノードの利用 ジョブ割当リソース

資源タイプ	資源タイプ名	説明	割り当て物理 CPUコア数	割り当て GPU数	メモリ(GiB)	ローカル ストレージ(GB)	資源タイプ 課金係数	割当可能な 個数
Full	rt_F	ノード占有	40	4	360	1440	1.00	1-512 (1-32)*
G.large	rt_G.large	ノード共有 GPU利用	20	4	240	720	0.90	1
G.small	rt_G.small	ノード共有 GPU利用	5	1	60	180	0.30	1
C.large	rt_C.large	ノード共有 CPUのみ利用	20	0	120	720	0.60	1
C.small	rt_C.small	ノード共有 CPUのみ利用	5	0	30	180	0.20	1
M.large	rt_M.large	ノード共有 CPUのみ利用	8	0	800	480	0.40	1
M.small	rt_M.small	ノード共有 CPUのみ利用	4	0	400	240	0.20	1

\* バッチジョブは最大512個、インタラクティブジョブは最大32個

資源タイプ名と数量で資源量を指定 例) 資源タイプ(Full)を1個指定: -I rt\_F=1

### 2. 計算リソース 制限事項

hour:minute:second 時間:分:秒

		インタラクティブ ジョブ (上限/デフォルト)	バッチジョブ (上限/デフォルト)	予約	
New!	rt_F		72:00:00/1:00:00	1日単位 最大30日間	
	rt_G.large rt_C.large	10.00.00 /1.00.00		NA	
	rt_M.large rt_M.small	12:00:00/1:00:00			
	rt_G.small rt_C.small		168:00:00/1:00:00		
	ノード時間積	12	2,304		

- 72時間(3日間)以上必要な学習は、「予約」を利用する。
- 小さなジョブは、一度に多数投入できる(分散処理向き)。



### 3. 計算ノードの予約利用

項目	説明
最小予約日数	1日
最大予約日数	30日
システムあたりの最大同時予約可能ノード数	442ノード
1予約あたりの最大予約ノード数 *システムでの最大同時予約数の制限があるため、最大 予約ノード数未満又はノードを予約できない場合がある。	32ノード
1予約あたりの最大予約ノード時間積	12, 288ノード時間積
予約受付開始時刻	30日前の午前10時
予約受付締切時刻	予約開始前日の午後9時
予約取消受付期間	予約開始前日の午後9時
予約開始時刻	予約開始日の午前10時
予約終了時刻	予約終了日の午前9時30分

予約ノードを利用する際は、qsubコマンド又はqrshコマンドにて、「-ar ar\_id」オプションを使う。

### 4. ABCI料金表: 2019/2020年度

		インタラクティブ ジョブ	バッチジョブ	予約
	rt_F	1.0 ポイント/時間		36ポイント/日 (1.5ポイント/時間)
	rt_G.large	0.9ポイント/時間		
計算資源	rt_C.large	0.6ポイン	<b>ル</b> /時間	
	rt_G.small 0.3ポイント/時間			NA
	rt_C.small 0.2ポイント/時間			
New! -	rt_M.large	0.4ポイント/時間		
	rt_M.small	rt_M.small 0.2ポイント/時間		
<b>711 3</b> 5	グループ領域 (共有ディスク)	5 ポイント/TB・月 (ABCIアカウント毎に割当られる200GBの		のホーム領域は無償)
ストレージ New!	ABCIクラウド ストレージ		目	



# 5. ジョブ実行上の特長大規模演算を必要とする場合、有効

- バッチジョブ(Spot)では、**最大512ノード**までを利用可能
- インタラクティブジョブ(On-demand)でも、最大32ノードまで利用可能
- ジョブ投入数(ユーザあたり)
  - 1) アレイジョブあたりの最大投入可能タスク数: **75,000**
  - 2) 同時投入可能ジョブ数: 1,000
  - 3) 同時実行ジョブ数: 200

### ABCI参考サイト

#### より詳細な情報については、以下を参照下さい。

- ・ ABCIユーザガイド https://docs.abci.ai/ja/
- ABCI利用ポータル
  <a href="https://portal.abci.ai/user/">https://portal.abci.ai/user/</a>
- ABCI利用ポータルの利用ガイド https://docs.abci.ai/portal/ja/
- ABCIユーザサポート
  <a href="https://abci.ai/ja/how\_to\_use/user\_support.html">https://abci.ai/ja/how\_to\_use/user\_support.html</a>
- ABCI利用法に関するFAQ https://abci.ai/ja/how\_to\_use/yakkan.html#faq9



