

# Hail-jp 「The 2nd」 Hands on seminars

第1回 Hail をインストールしてみよう！

**日時**  
2022年 1月 27 日

**会場**  
オンライン開催

mac でも windows でも Linux でも OK  
全3回開催のオンラインセミナーで、インタラクティブでスケーラブルな遺伝統計研究のためのライブラリ Hail をお手元の環境で試せるようになります

第1回 Hail をインストールしてみよう！(1/27 開催)  
第2回 GWAS tutorial を動かしてみよう！(2月下旬開催予定)  
第3回 Hail を使ってポリジェニックスコアを計算してみよう！(3月下旬開催予定)

お問い合わせは [hail-jp-staff@googlegroups.com](mailto:hail-jp-staff@googlegroups.com) まで

会場等協力：  
**GAJ**  
Genomic Analytics Japan  
**NABE International**  
株式会社 ナベインターナショナル



## Hail-jp ハンズオンセミナー

「Hailをインストールしてみよう！」

14:00- m1 mac

14:30- intel mac

15:00- Windows slot#1

15:30- Windows slot#2

16:00- Linux

解説：内田智博



# 今日の目標

---

自分の環境で動くhailを手に入れる！

できるだけプレーンな状態のマシンにインストールをデモします。

この組み合わせなら「動くはず」と「手順」を手に入れてください。



# 今日の流れ

---

## 1. (DEMO) 実機にインストール

並行してStep by Stepで実行していただきます

(適宜使用するソフトウェアのバージョンやポイントを解説します。)

## 2. Dockerを使ったHailを紹介

## 3. 質疑応答



# **1. (DEMO) 実機にインストール**

## **並行してStep by Stepで実行していただきます**

---

**注意!** まだほとんど使用していないPCへのインストールを前提としています。

使い込んでいる場合はすでにjavaやpythonがあるかもしれません。  
適宜読み替えていただくな、後ほどslackでご相談ください。



## 1. DEMO 実機にインストール

# Hailの動作要件

Windowsは公式にはサポートされていません

- 特に、Windowsに直接インストールしたpythonを使って構築していく場合は注意が必要です。
- 一見動くように見えますが、C:\¥を使ったファイルパスが災いして、エラーに見舞われます。

```
FatalError: IllegalArgumentException: Wrong FS: file:///C:/Users/tuchi/AppData/Local/Temp/tmpcfwsxk8i/1kg.vcf.bgz, expected: file:///  
Java stack trace:  
java.lang.IllegalArgumentException: Wrong FS: file:///C:/Users/tuchi/AppData/Local/Temp/tmpcfwsxk8i/1kg.vcf.bgz, expected: file:///  
    at org.apache.hadoop.fs.FileSystem.checkPath(FileSystem.java:730)  
    at org.apache.hadoop.fs.RawLocalFileSystem.pathToFile(RawLocalFileSystem.java:87)  
    at org.apache.hadoop.fs.RawLocalFileSystem.deprecatedGetFileStatus(RawLocalFileSystem.java:661)  
    at org.apache.hadoop.fs.RawLocalFileSystem.getFileLinkStatusInternal(RawLocalFileSystem.java:987)  
    at org.apache.hadoop.fs.RawLocalFileSystem.getFileStatus(RawLocalFileSystem.java:656)  
    at org.apache.hadoop.fs.FilterFileSystem.getFileStatus(FilterFileSystem.java:454)  
    at org.apache.hadoop.fs.ChecksumFileSystem$ChecksumFSInputChecker.<init>(ChecksumFileSystem.java:146)  
    at org.apache.hadoop.fs.ChecksumFileSystem.open(ChecksumFileSystem.java:347)  
    at org.apache.hadoop.fs.FileSystem.open(FileSystem.java:899)  
    at is.hail.io.fs.HadoopFS.openNoCompression(HadoopFS.scala:83)  
    at is.hail.io.fs.FS.copy(FS.scala:191)  
    at is.hail.io.fs.FS.copy$(FS.scala:190)  
    at is.hail.io.fs.HadoopFS.copy(HadoopFS.scala:70)
```

でも悲しむことはありません。 Windowsでも次のいずれかの方法をとればHailを動かすことができます。

- ① WSLを使ってWindows環境中にLinuxを起動しそのうちに構築する
- ② Docker Desktopを使用してDockerコンテナでHail環境を手に入れる

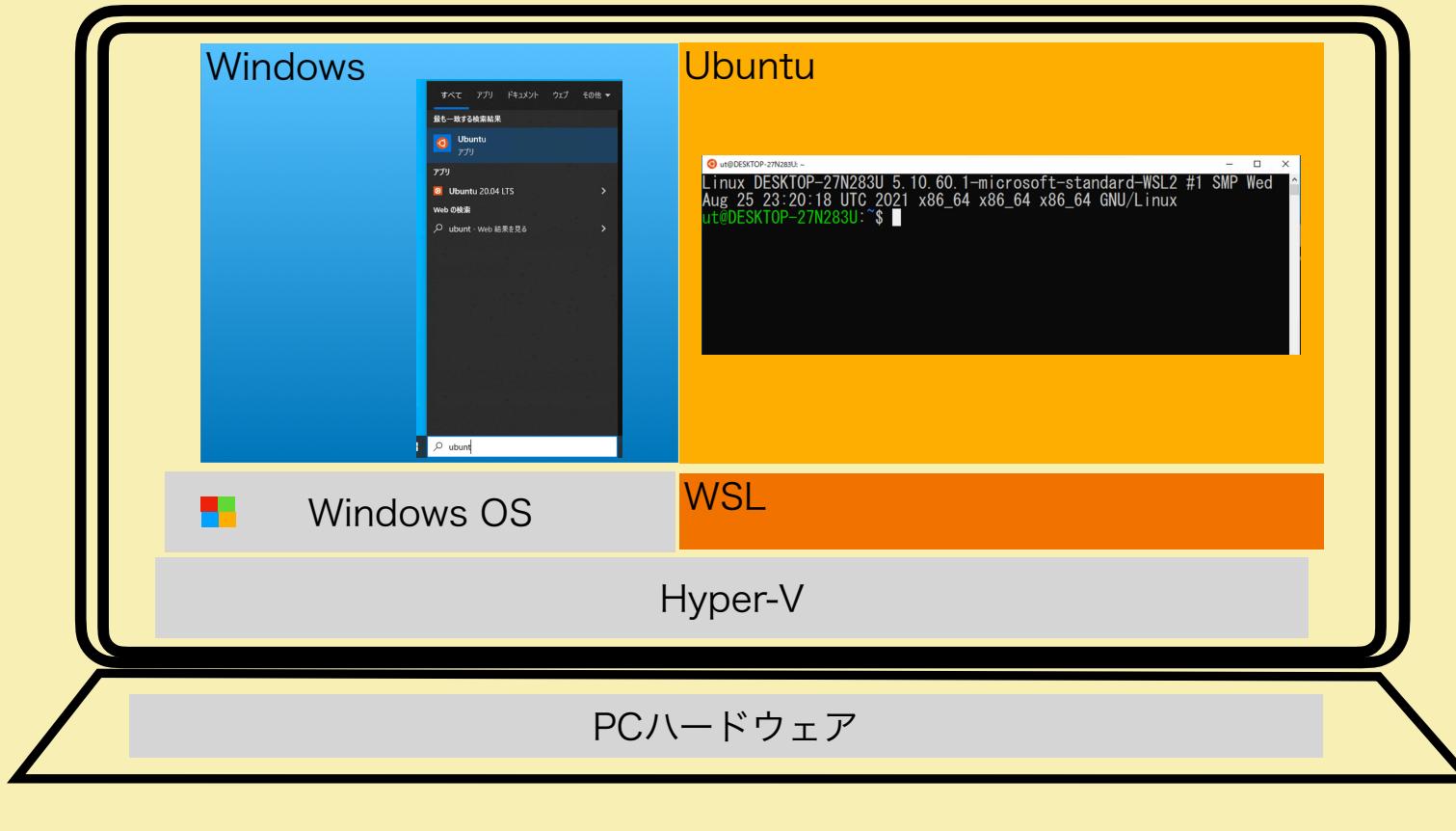


## 1. DEMO 実機にインストール

# WSLとは？？

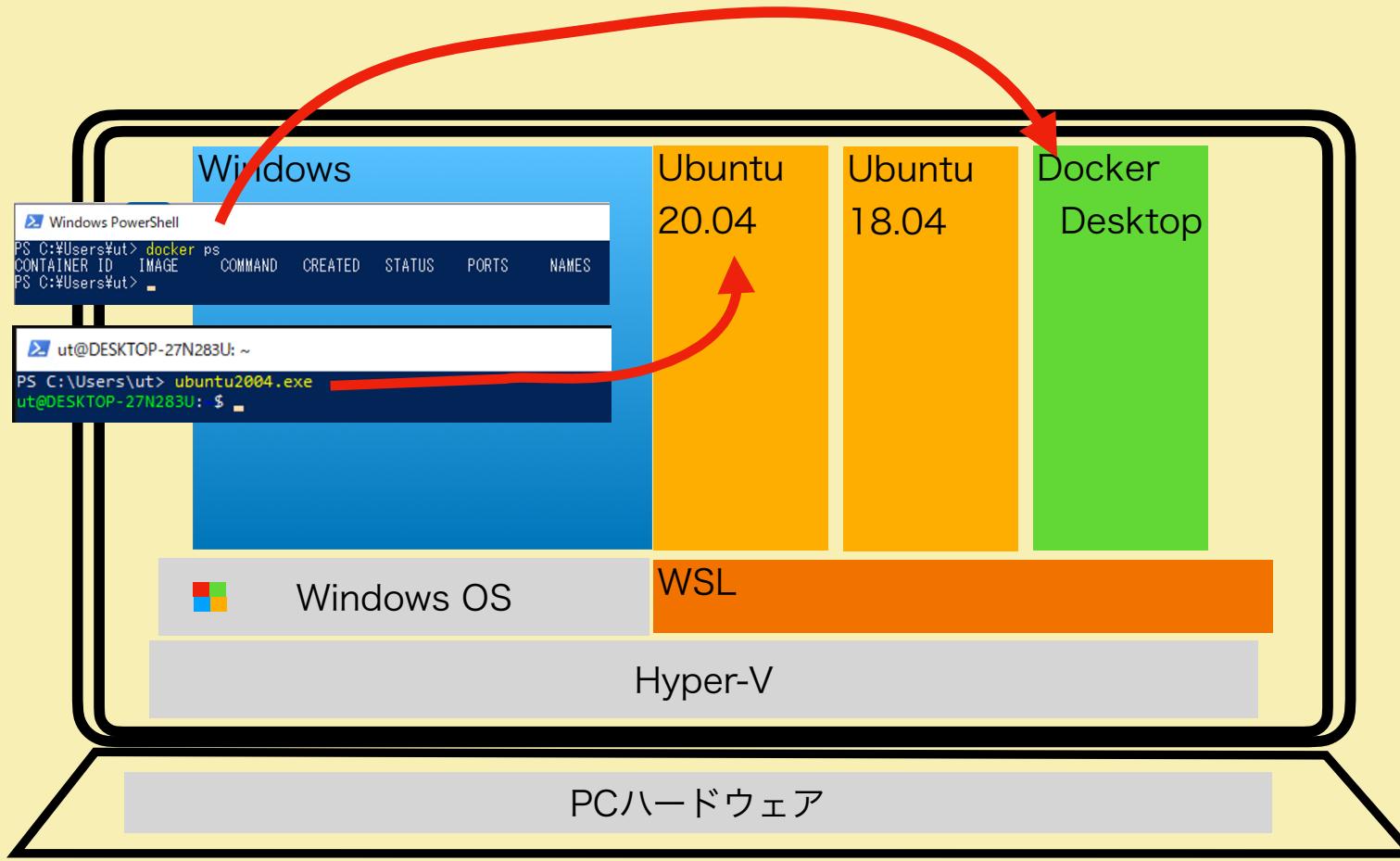
WSLは、「Windows Subsystem for Linux」のことです。WindowsのPCにLinuxを走らせることができるものです。いま現在、普通にインストールするとWSLはバージョン2が利用可能です。(以下、単にWSLと記すことにします)

WSLの仕組みを使うことで、Windows PCに簡単にLinux環境を導入することができるようになりました。



# Docker DesktopもWSLを使います

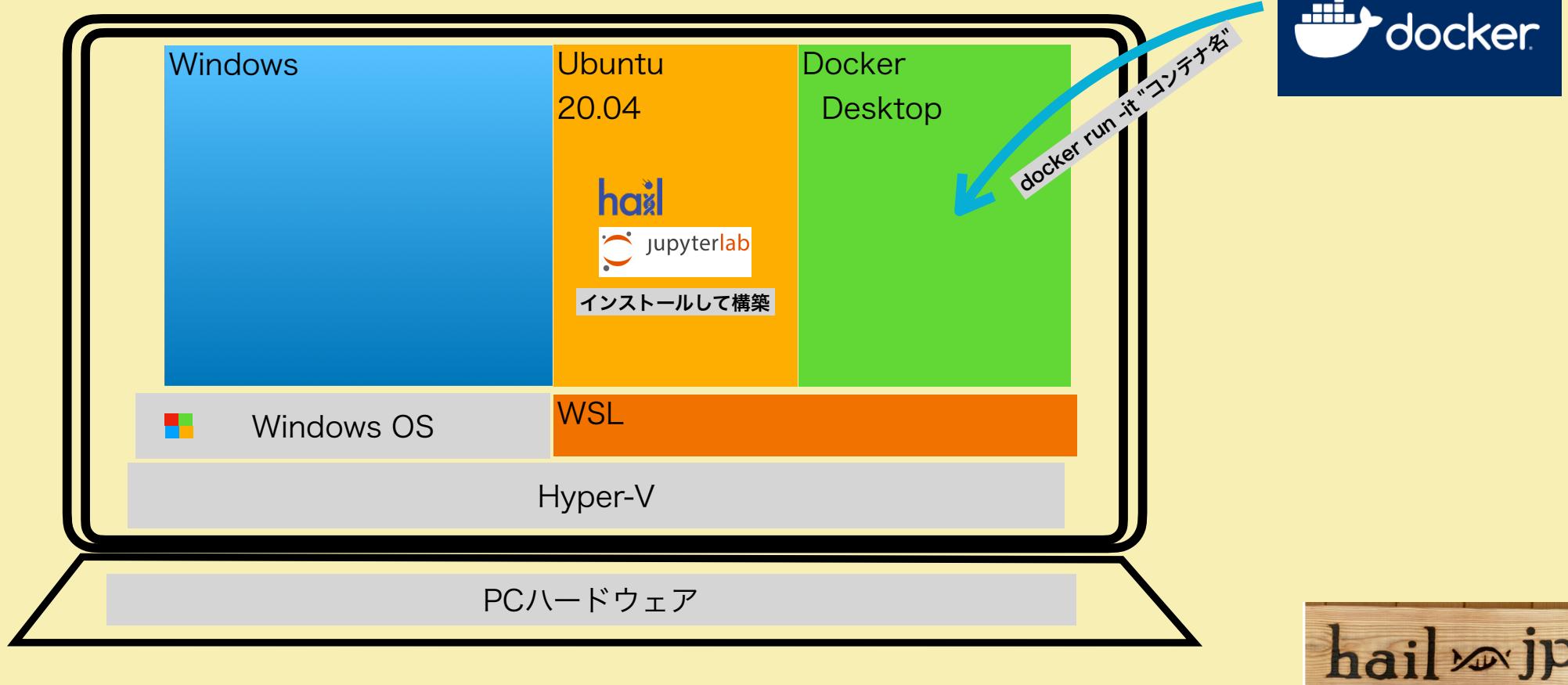
後述するDocker DesktopもWSLの仕組みを利用しています



## 1. DEMO 実機にインストール

# WSLやDocker Desktopを利用してHail環境入手

WindowsであってもWSLのおかげで、  
Ubuntu LinuxにHail環境を構築したり、  
Hail環境が入ったDockerコンテナを使うことができ、  
自分のHail環境を手にいれることができます。



## 1. DEMO 実機にインストール

# Hailの動作要件

Java 8 or 11

結局はLinux上への構築になりますので、RequirementはLinuxのものを参照します

(最近Java 11がサポートされました。今回は8で行きます)

## Install Hail on GNU/Linux

- Install Java 8 or Java 11.
- Install Python 3.7 or later.
- Install a recent version of the C and C++ standard libraries. GCC 5.0, LLVM version 3.4, or any later versions suffice.
- Install BLAS and LAPACK.
- Install Hail using pip.

Python

Hail

Jupyterlabも

<https://hail.is/docs/0.2/install/linux.html>



## 1. DEMO 実機にインストール

# Javaのバージョンを整理

Java 8 とは？？

Stable LTSとあるのが安定版

Javaの最新バージョンは18

Javaの最新の安定版は17

今回使うのは8

明示的に8を指定してインストールしないと、17が入ってしまい  
Hailが動かない場合があるので  
注意！

	Supported	Ready for Production	Role	Comment
JDK 8	+	✓	Stable LTS	Available as non-mainline 8u backport. Check with your vendor for availability. See known vendors list below.
JDK 9	!	✗		Discontinued, migrate to 11/17 <b>as soon as possible</b> .
JDK 10	!	✗		Discontinued, migrate to 11/17 <b>as soon as possible</b> .
JDK 11	✓	✓	Stable LTS	In mainline OpenJDK 11u <b>since 11.0.9</b> . Requires opt-in during build time, check with your vendor for availability. See known vendors list below.
JDK 12	!	✗		Discontinued, migrate to 17 <b>as soon as possible</b> .
JDK 13	!	!		Discontinued, migrate to 17
JDK 14	!	!	OpenJDK(オープンソースのJava) のホームページより	
JDK 15	!	!	Documentation, bug reports, etc... https://openjdk.java.net/jeps/	
JDK 16	!	!		Discontinued, migrate to 17.
JDK 17	✓	✓	Stable LTS	In mainline OpenJDK builds.
JDK 18	💡	💡	Dev/Test	In mainline OpenJDK builds.

<https://wiki.openjdk.java.net/display/shenandoah/Main>



# Java 8のバージョン

## 1. DEMO 実機にインストール

Java 8

セキュリティやbugfixのために  
できるだけ最新を使いたい

Java 8 の最新は 8u312

Java 8のrelease 8u312  
を使う!



<https://wiki.openjdk.java.net/display/jdk8u/Main>

Multi-Language VM

# OpenJDK Wiki

Dashboard > JDK 8u > Main

## Main

Created by Iris Clark, last modified by Andrew Hughes on Dec 28, 2021

## Welcome to OpenJDK 8 Updates!

OpenJDK 8 updates are a separate project of OpenJDK. Andrew Haley serves as the Project Lead. The list of Reviewers, Committee

## Maintainers

- Andrew Haley
- Andrew Hughes
- Severin Gehwolf

## Releases

Latest GA release: 8u312

Latest Generally Available (GA) binary releases of the OpenJDK jdk8u project are available at: <https://adoptopenjdk.net/upstream.html>

Latest Early Access (EA) binary releases of the OpenJDK jdk8u project are available at: <https://adoptopenjdk.net/upstream.html?var=ea>

### Most recent and past release details:

- 8u312-b07 (GA), October 19th 2021 [Release] [Tag] [Binaries]
- 8u302-b08 (GA), July 20th 2021 [Release] [Tag] [Binaries]
- 8u292-b10 (GA), April 20th 2021 [Release] [Tag] [Binaries]
- 8u282-b08 (GA), January 19th 2021 [Release] [Tag] [Binaries]
- 8u275-b01 (GA), November 5th 2020 [Release] [Tag] [Binaries]
- 8u272-b10 (GA), October 20th 2020 [Release] [Tag] [Binaries]

## 1. DEMO 実機にインストール

### 利用可能なJavaのビルドの種類と選択

Java 8

Azulu ?

複数のベンダーが  
OpenJDKをbuildして提供。  
それをインストールして使用する。

**Oracle Javaは利用条件要確認**

実際は  
WSL2上にあるUbuntu Linuxなので  
純正のopenjdkである、 **openjdk-8-jdk-headlessを選択**

ビルド	LTS	パーミッシブ	TCK	商用サポート
<a href="#">AdoptOpenJDK</a>	Yes	Yes	No	オプション(IBM)
<a href="#">Amazon Corretto</a>	Yes	Yes	Yes	オプション(AWS上)
<a href="#">Azul Zulu Builds of OpenJDK</a>	Yes	Yes	Yes	オプション
<a href="#">BellSoft Liberica JDK</a>	Yes	Yes	Yes	オプション
<a href="#">Eclipse Temurin</a>	Yes	Yes	Yes	オプション(Azul, IBM)
<a href="#">Microsoft Build of OpenJDK</a>	Yes	Yes	Yes	オプション(Azure上)
<a href="#">Oracle Java SE</a>	Yes	(バージョン次第)	Yes	Yes
<a href="#">Oracle OpenJDK</a>	No	Yes	Yes	No
<a href="#">Red Hat OpenJDK</a>	Yes	Yes	Yes	Yes
<a href="#">SapMachine</a>	Yes	Yes	Yes	オプション(SAP製品)

<https://wiki.openjdk.java.net/display/shenandoah/Main>



# pythonのバージョン

1. DEMO 実機にインストール

Python

hailの要求は3.7以上

Intel processor上で動くLinuxの場合はどれでもだいたい動きます

Pyrhonはubuntuのデフォルトを使用



# HailとJupyterlabのバージョン

1. DEMO 実機にインストール

Hail

Jupyterlab

ここまで選択を間違えなければ、これらは最新(=バージョン指定無し)でよい

Hailは最新を選択

Jupyterlabも最新を選択



# 今日行う構築のまとめ

## 1. DEMO 実機にインストール

Windows

③ ブラウザから動作確認

Webブラウザ

Ubuntu

① Ubuntu20.04上にJava8/Python/Hail/Jupyterlabをインストール

② jupyterlabを起動

最新のJupyterlab  
最新のHail

ubuntu標準のopenjdk

ubuntu標準のpython

ubuntu 20.04



Windows OS

WSL

④ WSLをインストールして動かす

Hyper-V

PCハードウェア

hail.jp

# では、インストールしていきましょう

## 1. DEMO 実機にインストール

今日打つコマンド一覧です。先にチャット欄に貼り付けます。

```
# Powershell上で、以下を実行し、WSL上のUbuntuを起動
```

```
Ubuntu
```

```
またはUbuntu2004
```

```
#Ubuntu上で、
```

```
## update
```

```
sudo apt update
```

```
## javaとpip3をインストール
```

```
sudo apt install openjdk-8-jre-headless python3-pip
```

```
## hailとjupyterlabをインストール
```

```
sudo pip3 install hail jupyterlab
```

```
## pythonとjavaのバージョン確認
```

```
python --version
```

```
java -version
```

```
## 動作確認 jupyterlabの起動
```

```
jupyter-lab --ip=0.0.0.0 --no-browser --NotebookApp.token="
```

```
# Windowsのブラウザで
```

```
http://localhost:8888
```

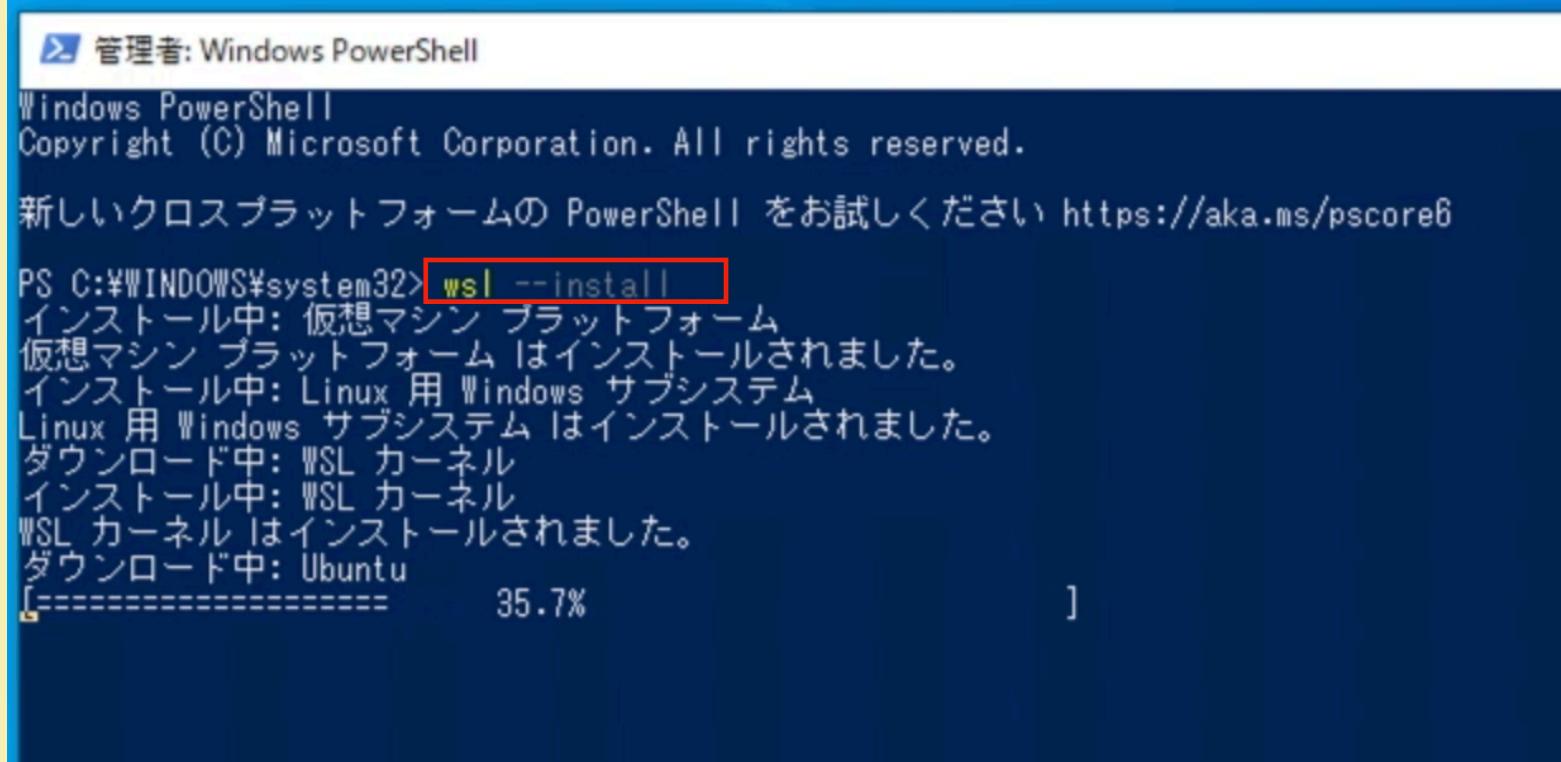
```
を開く
```



## 1. DEMO 実機にインストール

# WSLのインストール 1/3

管理者権限で開いた PowerShell で、コマンド 「wsl --install」 を実行するだけです。



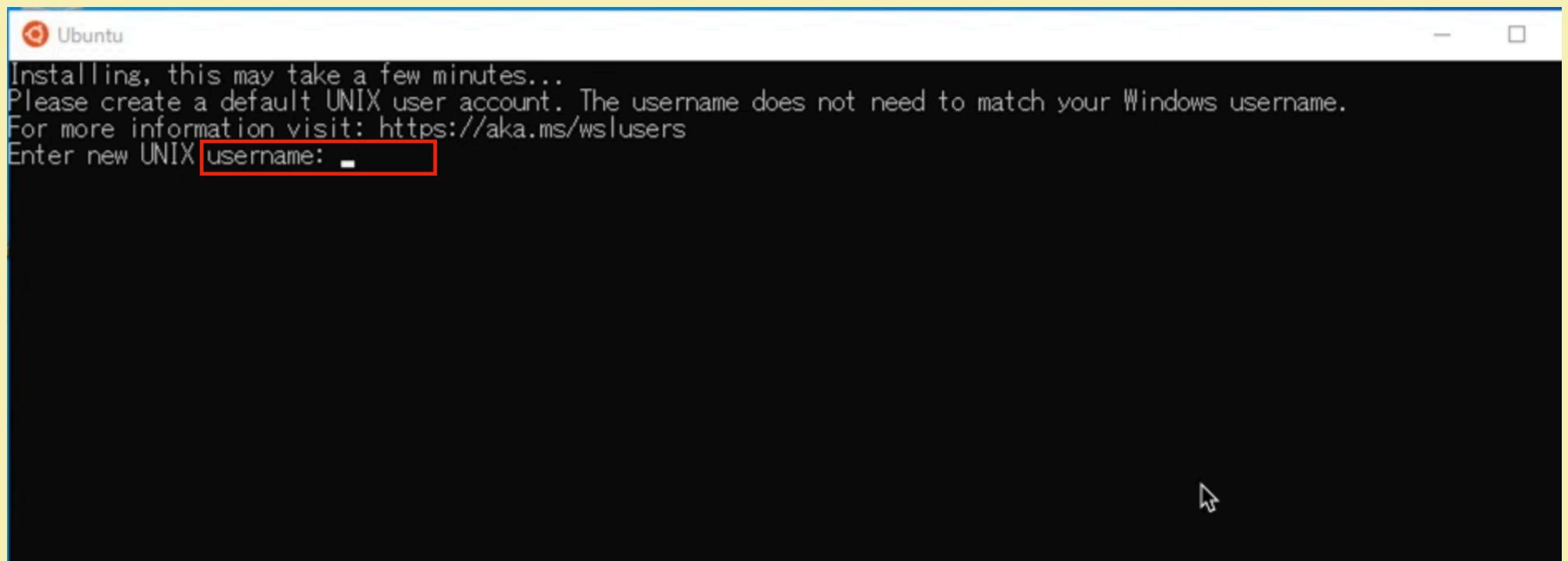
管理者: Windows PowerShell  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
新しいクロスプラットフォームの PowerShell をお試しください <https://aka.ms/pscore6>  
PS C:\WINDOWS\system32> **wsl --install**  
インストール中: フラットフォーム  
フラットフォーム はインストールされました。  
インストール中: Linux 用 Windows サブシステム  
Linux 用 Windows サブシステム はインストールされました。  
ダウンロード中: WSL カーネル  
インストール中: WSL カーネル  
WSL カーネル はインストールされました。  
ダウンロード中: Ubuntu  
[===== 35.7% ]

## 1. DEMO 実機にインストール

# WSLのインストール 2/3

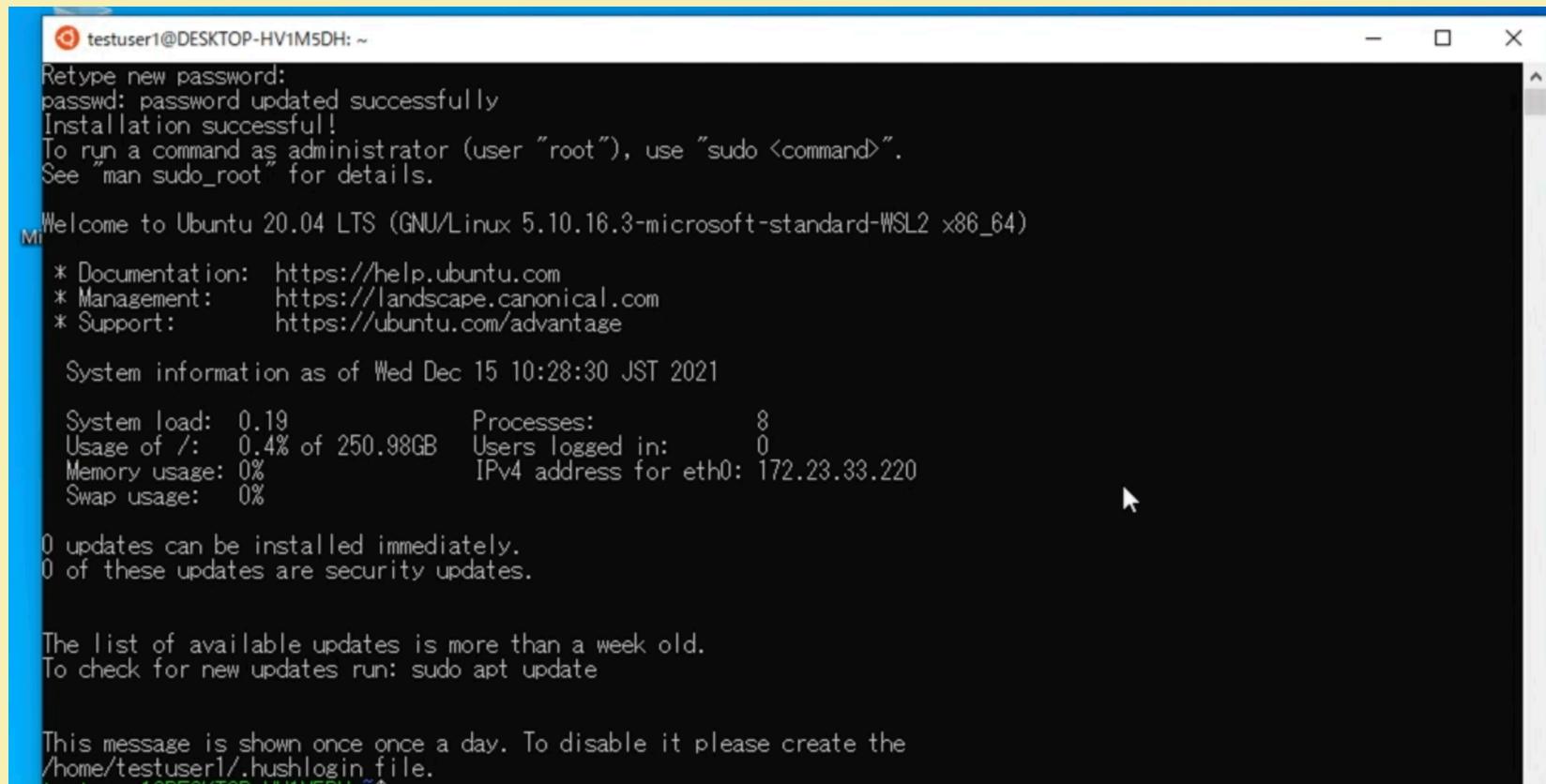
Windowsの導きに従い再起動をし、少々待つと、次のような画面が表示されます。

usernameとpasswordをいれて、このLinuxで使うユーザーを作ります。 Windowsのユーザーとの関連はありません。



# WSLのインストール 3/3

ユーザーの作成が終われば完了です。このコマンドラインはUbuntu Linuxの世界のものです。



```
testuser1@DESKTOP-HV1M5DH: ~
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

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

System information as of Wed Dec 15 10:28:30 JST 2021

System load: 0.19      Processes:          8
Usage of /: 0.4% of 250.98GB  Users logged in:  0
Memory usage: 0%          IPv4 address for eth0: 172.23.33.220
Swap usage:  0%

0 updates can be installed immediately.
0 of these updates are security updates.

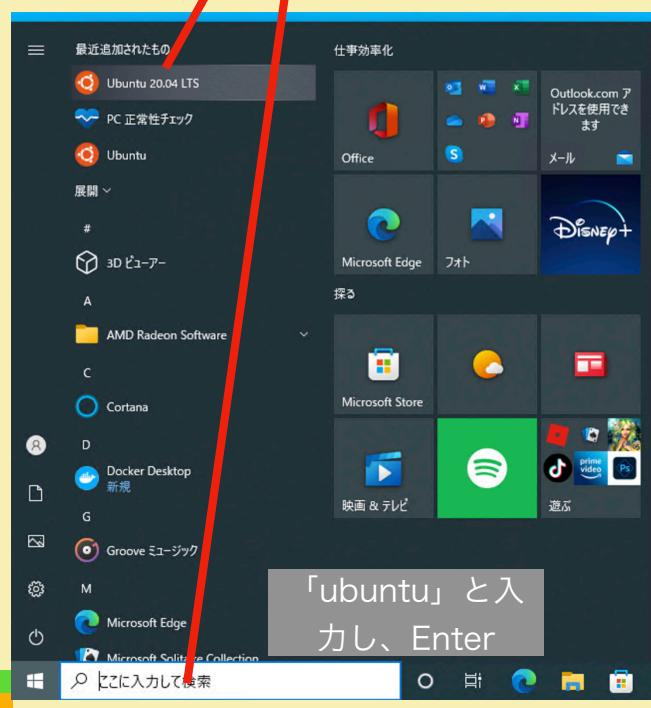
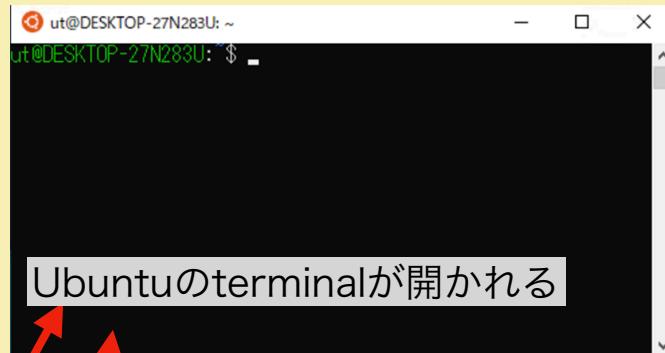
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once once a day. To disable it please create the
/home/testuser1/.hushlogin file.
testuser1@DESKTOP-HV1M5DH:~$
```

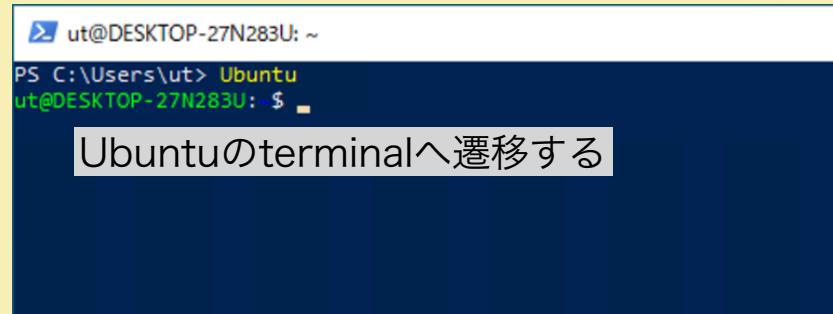


# Ubuntuの起動方法

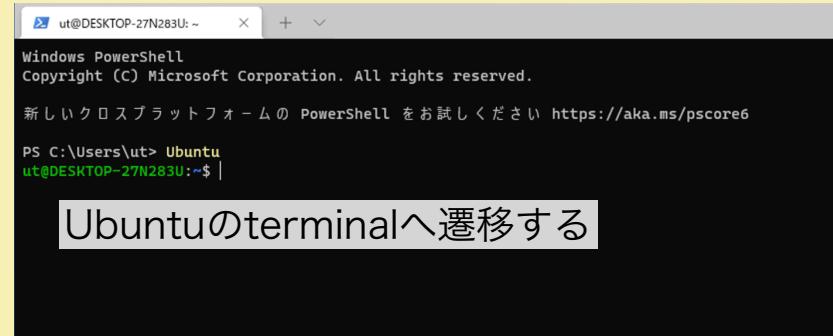
## 1. DEMO 実機にインストール



### Powershellを開き、Ubuntuを実行



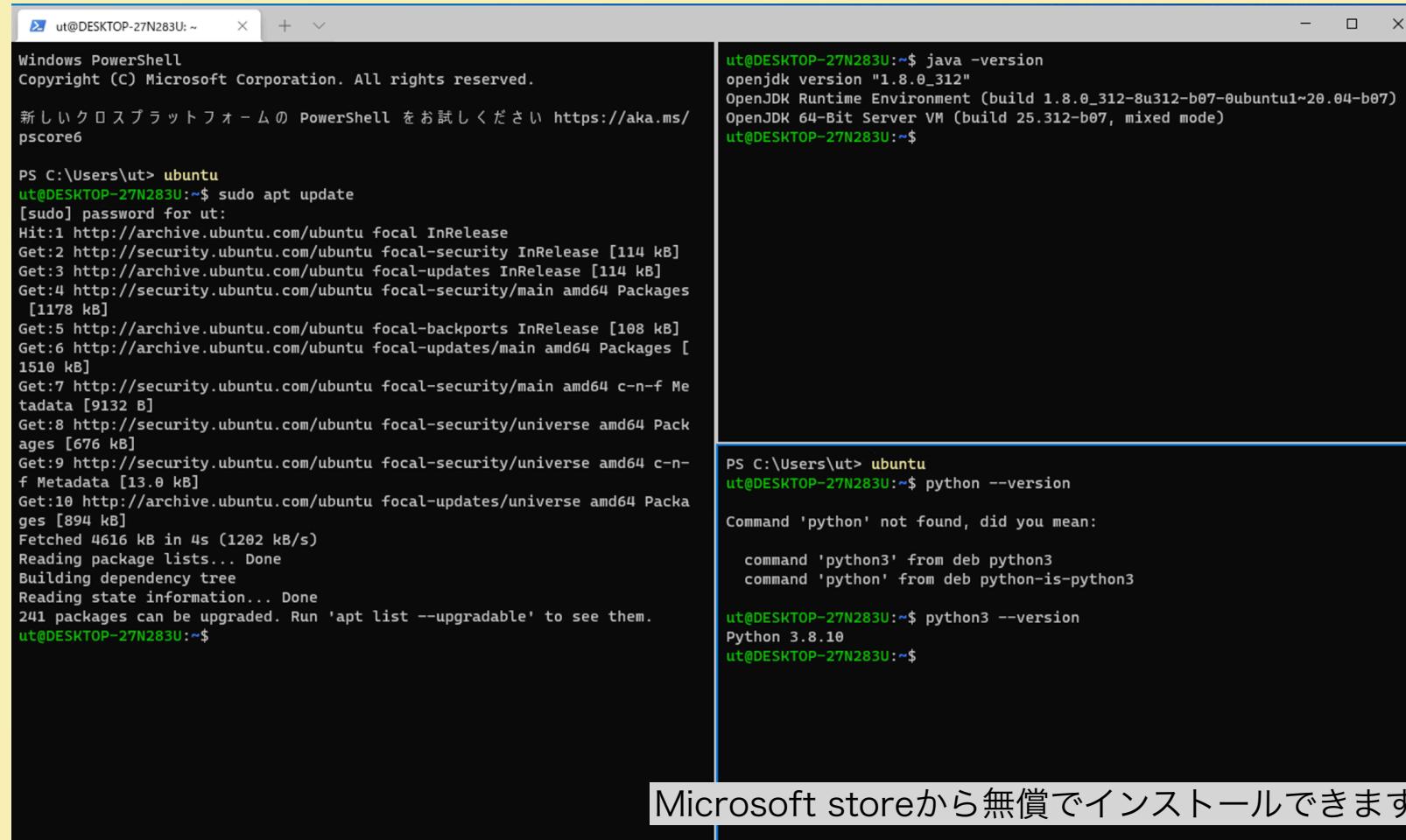
### Windows terminal※ を開き、Ubuntuを実行



# Windows terminalも便利そうです

## 1. DEMO 実機にインストール

このように、画面分割やタブ機能が利用できる、近代的なターミナルのようです



```
ut@DESKTOP-27N283U:~$ java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~20.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)

ut@DESKTOP-27N283U:~$ sudo apt update
[sudo] password for ut:
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1178 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1510 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [9132 B]
Get:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [676 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [13.0 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [894 kB]
Fetched 4616 kB in 4s (1202 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
241 packages can be upgraded. Run 'apt list --upgradable' to see them.
ut@DESKTOP-27N283U:~$
```

```
PS C:\Users\ut> ubuntu
ut@DESKTOP-27N283U:~$ python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3

ut@DESKTOP-27N283U:~$ python3 --version
Python 3.8.10
ut@DESKTOP-27N283U:~$
```

Microsoft storeから無償でインストールできます。

hail⚡jp

# ubuntu上へhail環境をインストール

## 1. DEMO 実機にインストール

ubuntuのパッケージ管理ツールをupdate

```
ut@DESKTOP-27N283U: ~$ sudo apt update
```

java8と、python3のモジュールをインストールするのに使うpip3をインストールします

```
ut@DESKTOP-27N283U: ~$ sudo apt install openjdk-8-jre-headless python3-pip
```

python3とjavaのバージョンを確認します

```
ut@DESKTOP-27N283U: ~$ python3 --version
Python 3.8.10
ut@DESKTOP-27N283U: ~$ java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~20.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
```

java8と、python3のモジュールをインストールするのに使うpip3をインストールします

```
ut@DESKTOP-27N283U: ~$ sudo pip3 install hail jupyterlab
```



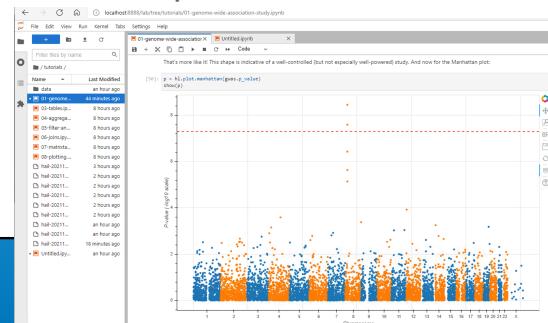
## 動作確認をしてみましょう

### 1. DEMO 実機にインストール

Ubuntu上に走らせたjupyter labへ、Windows側のブラウザからアクセスします

Windows

webブラウザで  
JupyterLabにアクセス  
<http://localhost:8888>  
notebookで"import hail as hl"



Ubuntu

terminalでJupyterLabを起動

```
jupyter-lab \
--ip=0.0.0.0 \
--no-browser \
--NotebookApp.token=""
```



Hyper-V

Windows OS

PCハードウェア

hail.jp

## 1. DEMO 実機にインストール

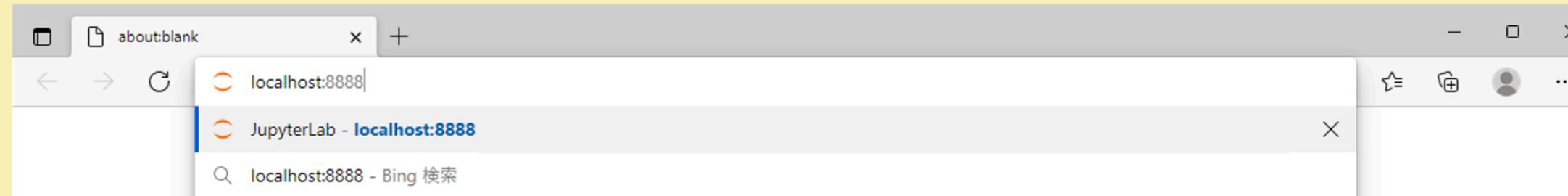
動作確認をしてみましょう

ubuntuで、jupyter-labを起動

jupyter-lab --ip=0.0.0.0 --no-browser --NotebookApp.token=""

```
ut@DESKTOP-27N283U:~$ jupyter-lab --ip=0.0.0.0 --no-browser --NotebookApp.token=''  
[W 2022-01-21 15:04:49.247 LabApp] 'token' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.  
[I 2022-01-21 15:04:49.253 ServerApp] jupyterlab | extension was successfully linked.  
[I 2022-01-21 15:04:49.585 ServerApp] nbclassic | extension was successfully linked.  
[W 2022-01-21 15:04:49.615 ServerApp] All authentication is disabled. Anyone who can connect to this server will be able to run code.  
[I 2022-01-21 15:04:49.623 ServerApp] nbclassic | extension was successfully loaded.  
[I 2022-01-21 15:04:49.624 LabApp] JupyterLab extension loaded from /usr/local/lib/python3.8/dist-packages/jupyterlab  
[I 2022-01-21 15:04:49.624 LabApp] JupyterLab application directory is /usr/local/share/jupyter/lab  
[I 2022-01-21 15:04:49.627 ServerApp] jupyterlab | extension was successfully loaded.  
[I 2022-01-21 15:04:49.628 ServerApp] Serving notebooks from local directory: /home/ut  
[I 2022-01-21 15:04:49.628 ServerApp] Jupyter Server 1.13.3 is running at:  
[I 2022-01-21 15:04:49.628 ServerApp] http://DESKTOP-27N283U:8888/lab  
[I 2022-01-21 15:04:49.628 ServerApp] or http://127.0.0.1:8888/lab  
[I 2022-01-21 15:04:49.628 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Windows上のウェブブラウザで、localhost:8888 を打ち込み、Enter

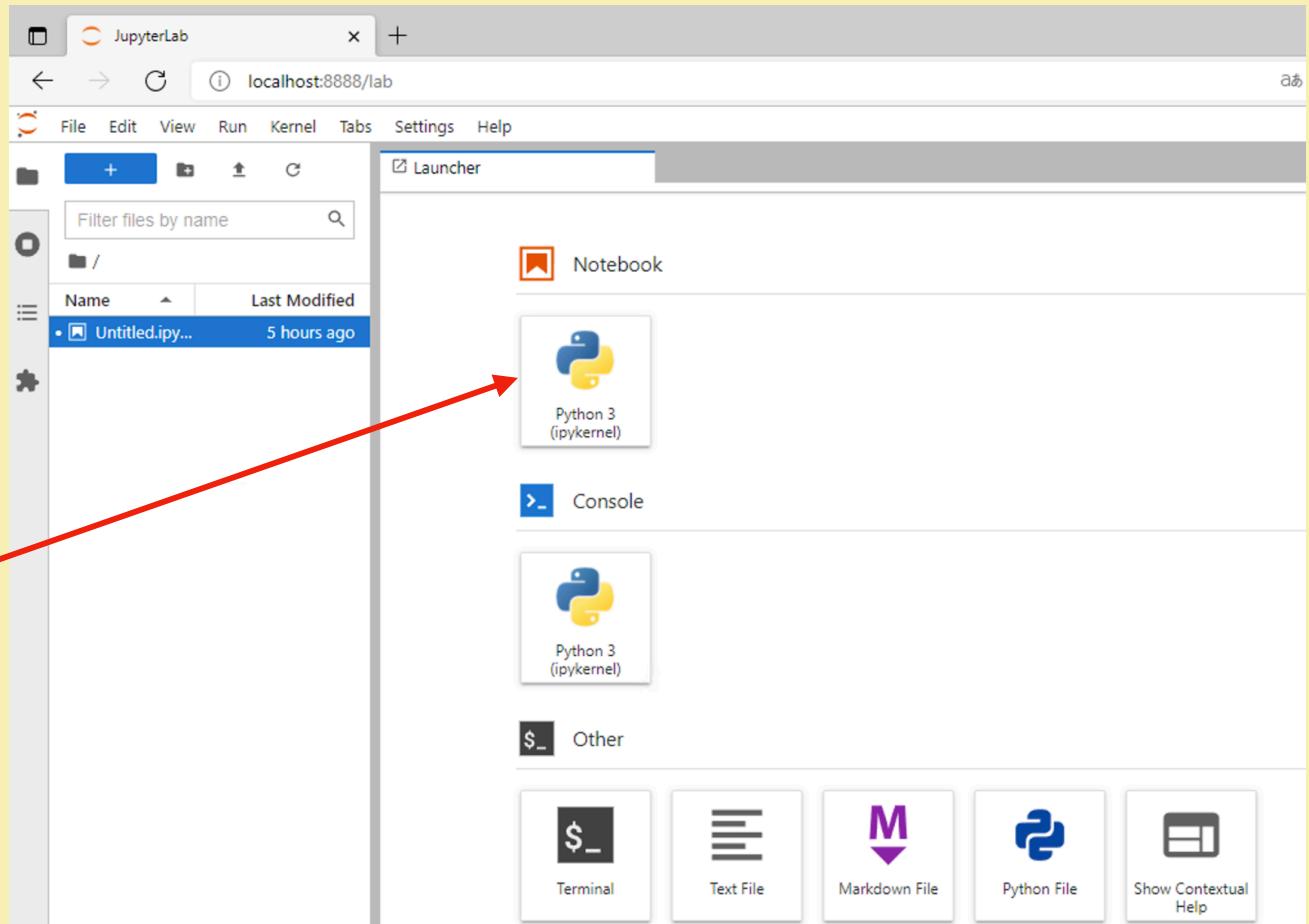


動作確認をしてみましょう

## 1. DEMO 実機にインストール

このような画面が出てきたらOK

Python3をクリックすると  
「Python kernel」が起動します



動作確認をしてみましょう

## 1. DEMO 実機にインストール

Python kernelが起動したら、

import hail as hl

を入力して「Shift + Enter」

次に、

hl.init()

を入力して「Shift + Enter」

を押下します。

このように、「Welcome to」や

Hailのステキなアスキーアートが出力されたらOKです！

The screenshot shows a Jupyter Notebook interface with a single cell containing Python code. The code consists of two lines: [1]: `import hail as hl` and [2]: `hl.init()`. The output of the cell is a series of log messages and a decorative ASCII art logo. The log messages include warnings about hostname resolution and Spark configuration, followed by a welcome message and the Hail logo. The logo is a stylized tree or forest made of slashes.

```
[1]: import hail as hl
[2]: hl.init()
2022-01-21 15:20:42 WARN  Utils:69 - Your hostname, DESKTOP-27N283U resolves to a loopback address instead of 172.28.89.162 (on interface eth0)
2022-01-21 15:20:42 WARN  Utils:69 - Set SPARK_LOCAL_IP if you need to bind to another address
2022-01-21 15:20:43 WARN  NativeCodeLoader:60 - Unable to load native-hadoop library for your platform built-in Java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Running on Apache Spark version 3.1.2
SparkUI available at http://172.28.89.162:4040
Welcome to
   _/\_ _/\_ _/\_ _/\_
  / \ / \ / \ / \ / \
 / / / \ \ / \ / \ / \
version 0.2.81-edeb70bc789c
LOGGING: writing to /home/ut/hail-20220121-1520-0.2.81-edeb70bc789c.log
```



## 1. DEMO 実機にインストール

### (参考) WindowsとWSL2上とのファイルの関係

Windows

WSLで動かしているLinuxのファイルシステムは、  
Windows上ではひとつのファイル(ext4.vhdx)として存在しています

Ubuntu

ext4.vhdx  
WindowsのDISK

/usr/bin/java  
/usr/bin/python3  
/home/ut/hail.ipynb  
...  
Linux  
Filesystem

Hyper-V



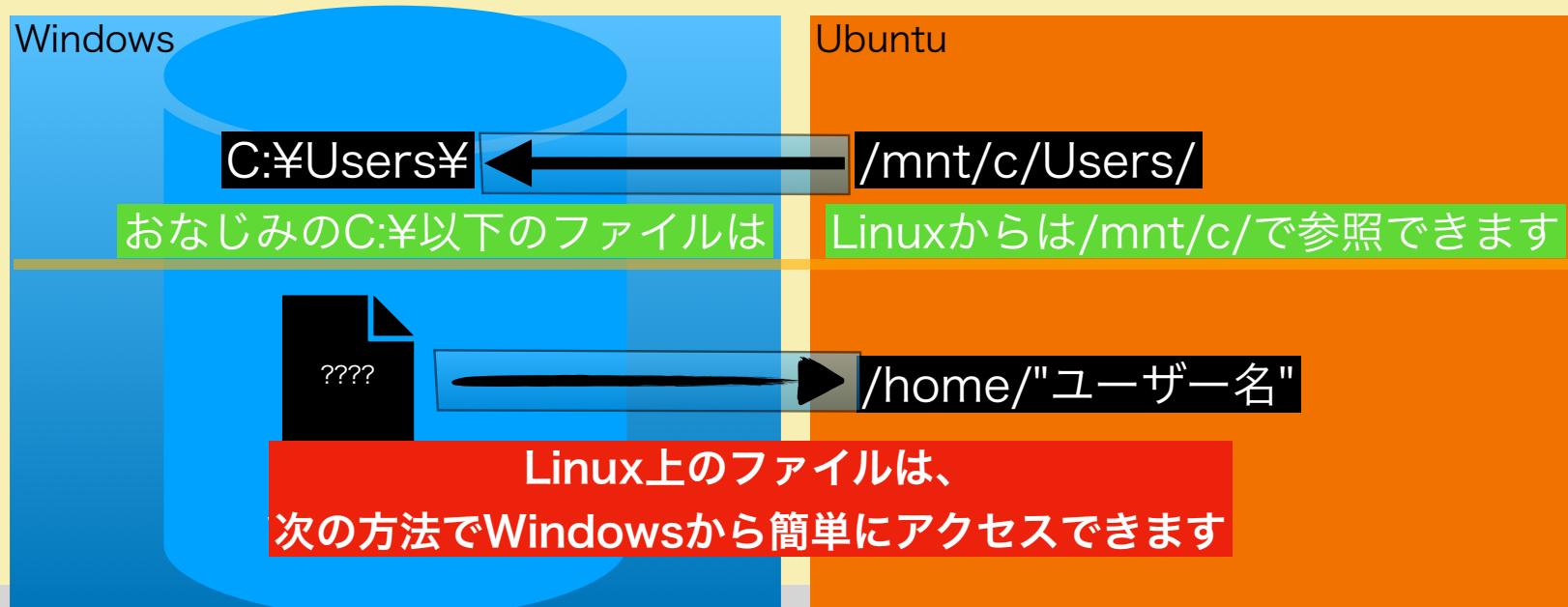
Windows OS

PCハードウェア

hail.jp

## (参考) WindowsとWSL2上とのファイルの関係

### 1. DEMO 実機にインストール



Windows OS

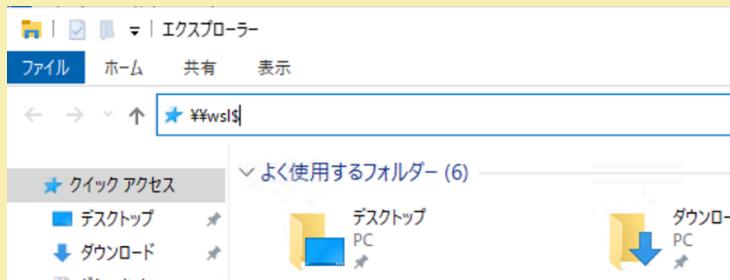
PCハードウェア

hail jp

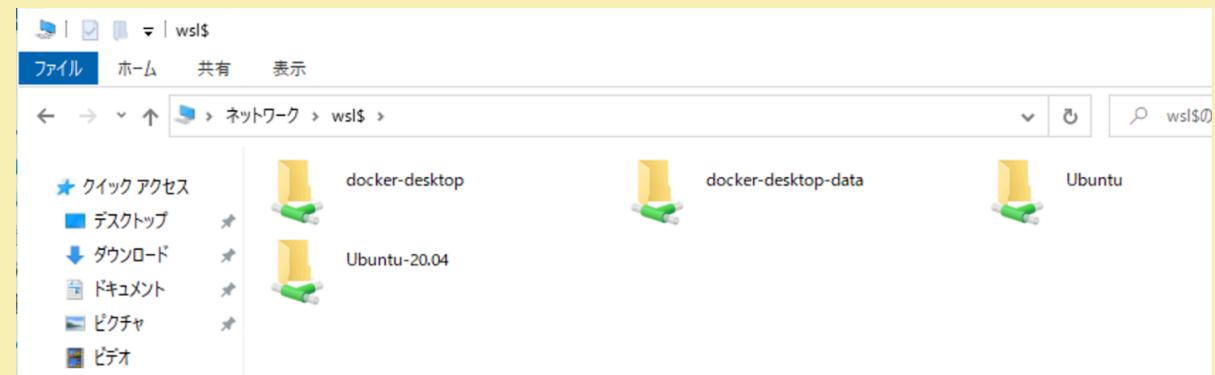
## (参考) WSL上のubuntuにあるファイルの参照

### 1. DEMO 実機にインストール

エクスプローラーのアドレスバーに  
「¥¥wsl\$」  
を入力してEnter すると・・・



WSLで動かしているLinux (たち)の名前が表示されます



中に入っていくと、ubuntuの/home以下なども参照できます



## 2. Dockerを使ったHailを紹介

---



# WindowsでDockerコンテナを利用するには？？

2. Dockerを使ったHailの紹介

標準的には、Docker Desktopが必要です

2021年8月31日にライセンス改訂があり、無償で使用できていたDocker Desktopは  
「従業員数250人以上や年間収益1,000万ドル以上の企業」の場合は有料プランへの加入が必要になりました。

その場合はどのプランが必要？ -> 「Docker Pro \$5/month」以上のプランです

- Our [Docker Subscription Service Agreement](#) includes a change to the terms for Docker Desktop
- It **remains free** for small businesses (fewer than 250 employees AND less than \$10 million in annual revenue), personal use, education, and non-commercial open source projects.
  - It requires a paid subscription ([Pro, Team or Business](#)), for as little as \$5 per user per month, for professional use in larger businesses.
  - The effective date of these terms is August 31, 2021. **There is a grace period until January 31, 2022** for those that will require a paid subscription to use Docker Desktop.
  - The Docker Pro, Docker Team, and Docker Business subscriptions **now include** commercial use of Docker Desktop.
  - Check out our [FAQ](#) for more information. Or read our [latest blog](#).

small businessのほかは、  
personal use, education,  
non-commercial opensource project  
がfreeのままであります。

FAQにはより具体的なケースについて解説があります。

この条件は、2022年1月31日までの猶予期間があるので、今日(1/27)は無償で使えます。



## Docker Desktopを利用してhail環境を得る

Docker desktopをインストール

Docker desktopのwebsiteからWindows用のパッケージをダウンロードしてインストール

Powershellなどでdockerコマンドが利用可能になります

Docker desktopがインストールできたら

```
docker run -it -p 8888:8888 -p 4040:4040 utut/hail-jp-trial:latest
```

( -p 8888:8888はjupyterlabのポートを転送, -p 4040:4040はSparkUIのポートを転送しています)

utut/hail-jp-trial:latestは今回のために作った最小限のhail環境です

ブラウザにhttp://localhost:8888 と打ち込めばjupyterlabを開けるようになります



# Docker DesktopをWindowsへインストール

2. Dockerを使ったHailの紹介

- Docker Desktopのwebページを開き、インストーラーをダウンロード



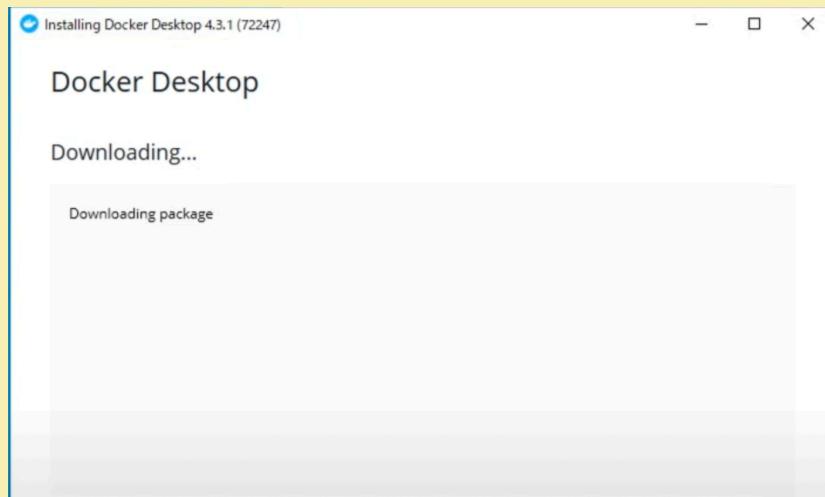
<https://www.docker.com/products/docker-desktop>



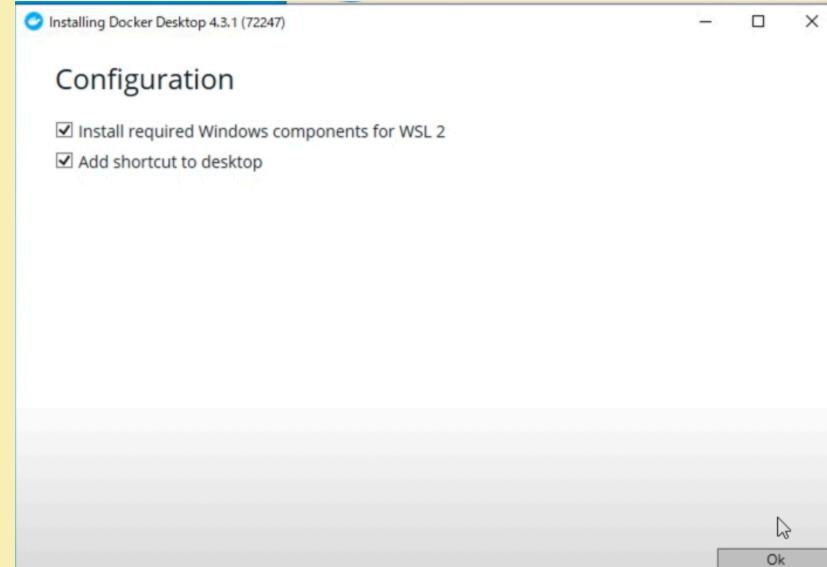
# Docker DesktopをWindowsへインストール

2. Dockerを使ったHailの紹介

Downloadしたインストーラーを実行し、  
インストールをすすめていきます



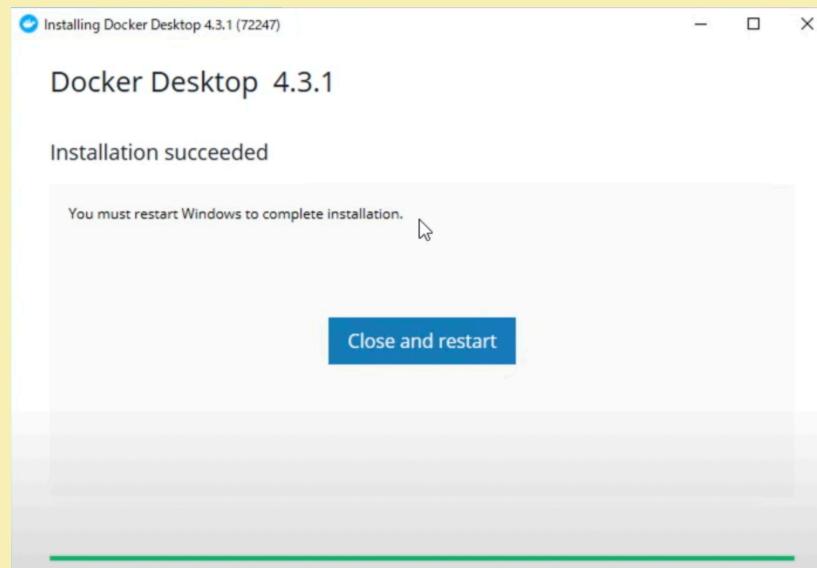
ConfigurationもそのままOK



# Docker DesktopをWindowsへインストール

## 2. Dockerを使ったHailの紹介

再起動を促されます



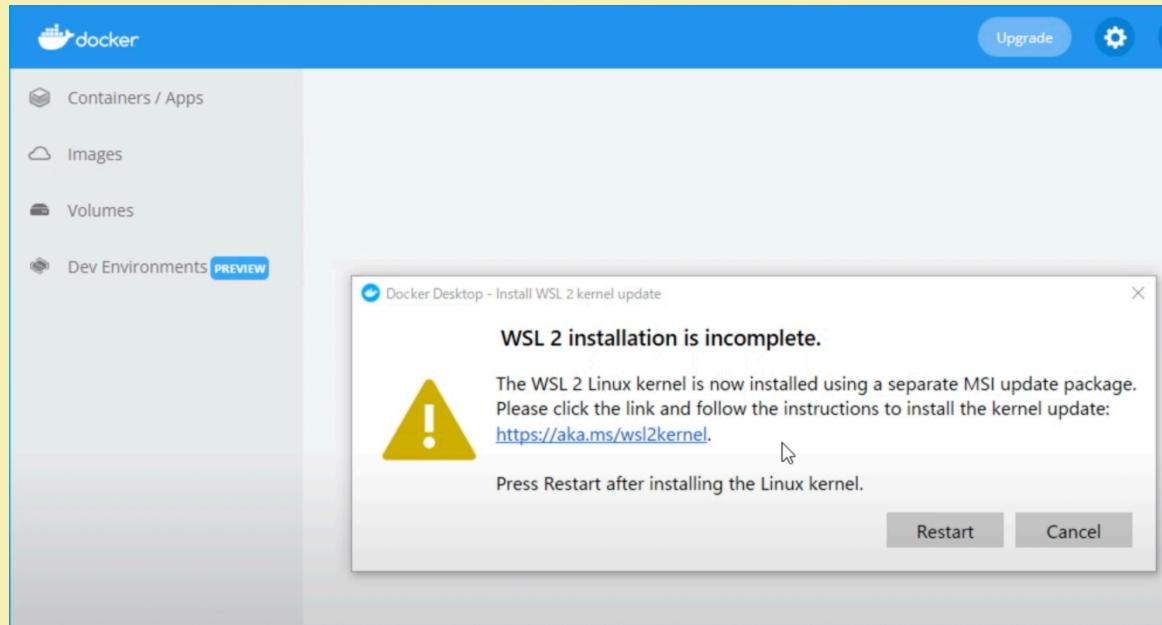
再起動後、ライセンスの確認があります



# Docker DesktopをWindowsへインストール

(先にWSL環境でubuntuを動かしたりしていると出ません)

リンクの先にあるPackageをインストールすることを促されます



リンクを辿って指示に従っても解決できますが、

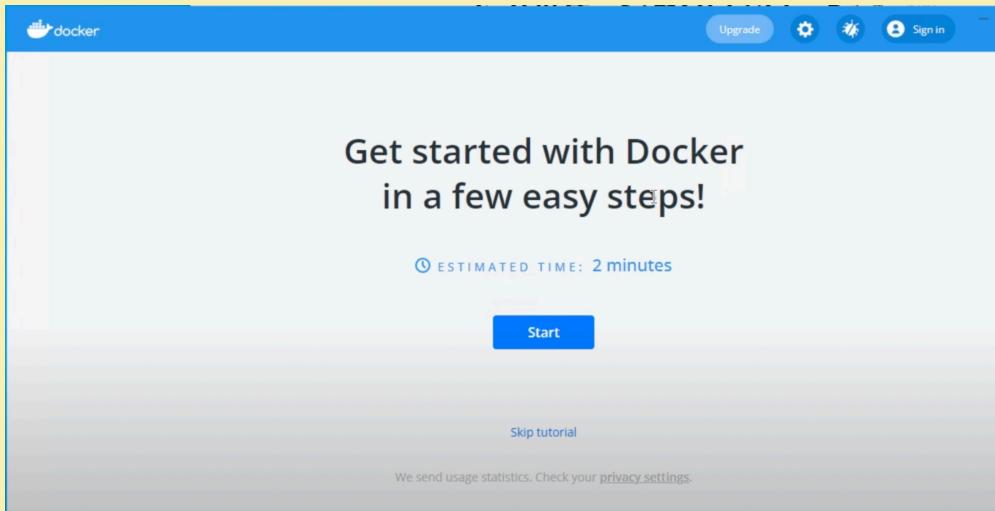
管理者権限でPowerShellを開き、  
wsl --update  
を実行すると解決できます。



# Docker DesktopをWindowsへインストール

2. Dockerを使ったHailの紹介

この画面が出てくればOK！



PowerShellで「`docker ps`」が実行可能になっているはずです

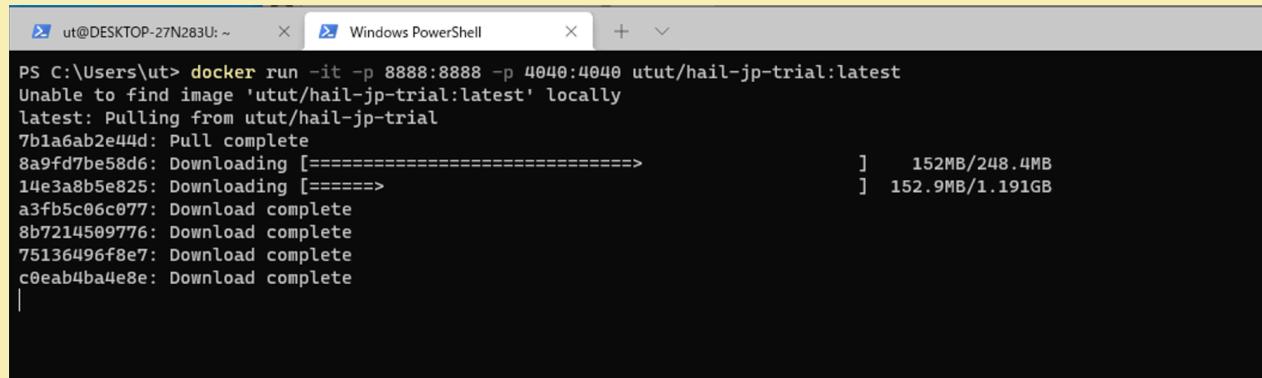
A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command "PS C:\\$Users\\$ut> docker ps" is run, displaying a table of container information. The columns are: CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, PORTS, and NAMES. There is one row of data shown. The PowerShell window has a dark blue background.

## 2. Dockerを使ったHailの紹介

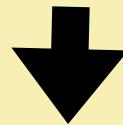
動作確認をしてみましょう

jupyterlabを起動して

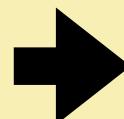
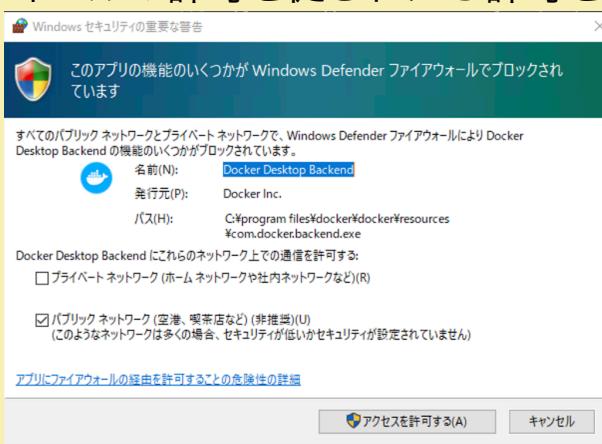
docker run -it -p 8888:8888 -p 4040:4040 utut/hail-jp-trial:latest



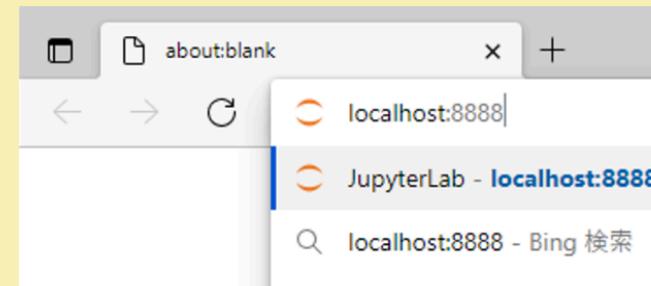
```
PS C:\Users\ut> docker run -it -p 8888:8888 -p 4040:4040 utut/hail-jp-trial:latest
Unable to find image 'utut/hail-jp-trial:latest' locally
latest: Pulling from utut/hail-jp-trial
7b1a6ab2e44d: Pull complete
8a9fd7be58d6: Downloading [=====]
14e3a8b5e825: Downloading [=====]
a3fb5c06c077: Download complete
8b7214509776: Download complete
75136496f8e7: Download complete
c0eab4ba4e8e: Download complete
```



ファイアウォールの許可を促されたら許可をします



ブラウザに、<http://localhost:8888>を貼り付けます

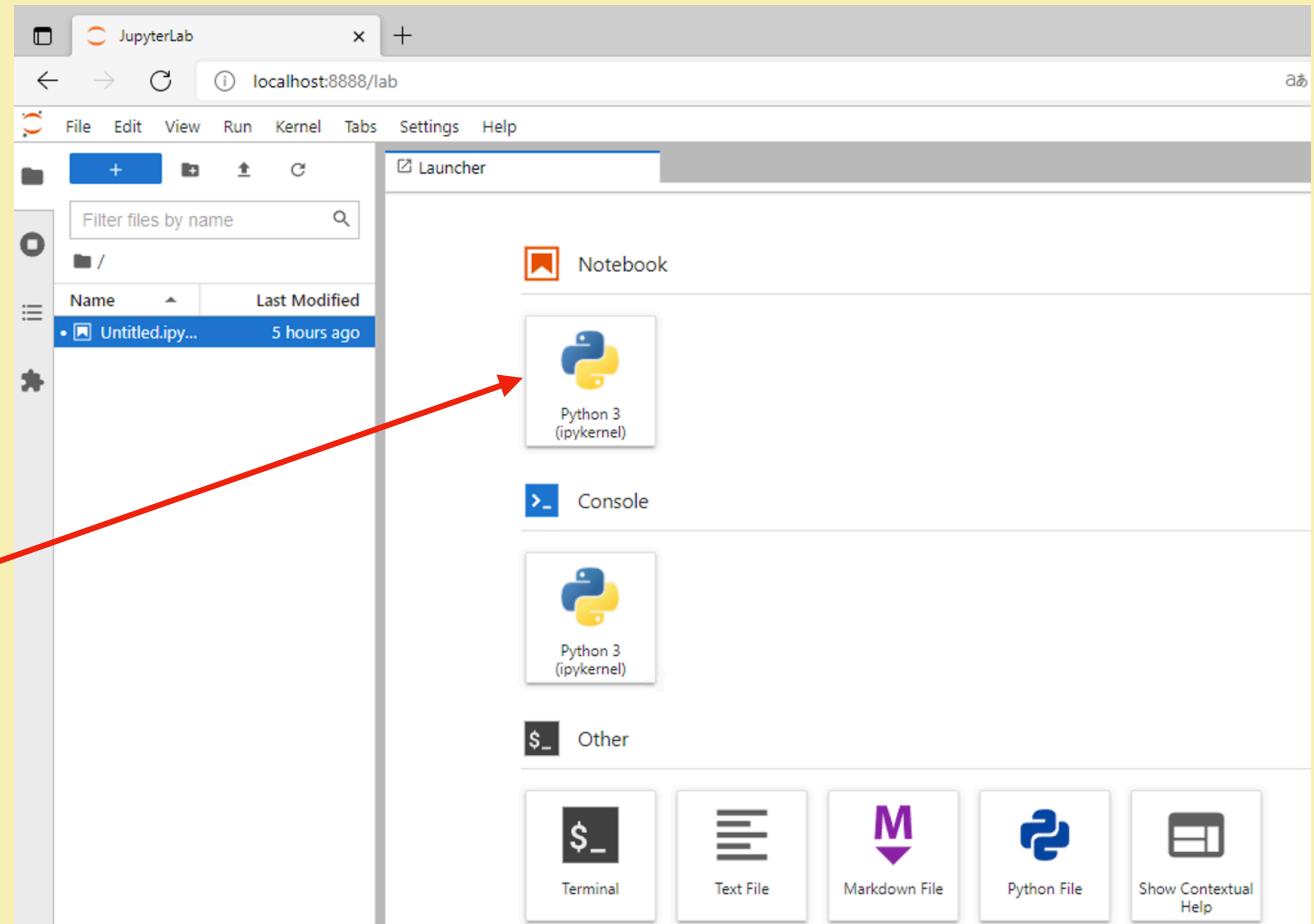


## 2. Dockerを使ったHailの紹介

動作確認をしてみましょう

このような画面が出てきたらOK

Python3をクリックすると  
「Python kernel」が起動します



動作確認をしてみましょう

Python kernelが起動したら、

import hail as hl  
を入力して「Shift + Enter」

次に、  
hl.init()  
を入力して「Shift + Enter」  
を押下します。

このように、「Welcome to」や  
Hailのステキなアスキーアートが出力されたらOKです！

The screenshot shows a Jupyter Notebook interface with a single cell titled 'Untitled1.ipynb'. The cell contains two lines of Python code: '[1]: import hail as hl' and '[2]: hl.init()'. The output of line [2] is a series of log messages and a decorative ASCII art logo. The log messages include warnings about hostname resolution and Spark configuration, followed by a 'Welcome to' message and the Hail version information ('version 0.2.81-edeb70bc789c'). A red arrow points from the explanatory text on the left to the cell's input line, and another red arrow points from the explanatory text to the cell's output.

```
[1]: import hail as hl
[2]: hl.init()
2022-01-21 15:20:42 WARN  Utils:69 - Your hostname, DESKTOP-27N283U resolves to a loopback address
using 172.28.89.162 instead (on interface eth0)
2022-01-21 15:20:42 WARN  Utils:69 - Set SPARK_LOCAL_IP if you need to bind to another address
2022-01-21 15:20:43 WARN  NativeCodeLoader:60 - Unable to load native-hadoop library for your platform
builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Running on Apache Spark version 3.1.2
SparkUI available at http://172.28.89.162:4040
Welcome to
   _/\_ _/\_ _/\_ _/\_
  / \ / \ / \ / \ / \
 / / / \ \ / \ / \ / \
version 0.2.81-edeb70bc789c
LOGGING: writing to /home/ut/hail-20220121-1520-0.2.81-edeb70bc789c.log
```

### 3. 質疑応答

宜しければ、Hail-jp のslackでもご相談ください



### 3. 質疑応答

宜しければ、Hail-jp のslackでもご相談ください



次回案内

hail ~ jp



WSL/WSL2を使うとWindows OSの中に仮想的なLinuxマシンを起動することができます。この分野、ずいぶん以前からそのようなものがありました。古くはCygwin、仮想化技術の進展に伴い、VM WareやVirtualboxなどで仮想的なx86マシンをつくり、Linuxをインストールすることができるようになりました。それが今やWindows謹製の環境でUbuntu LinuxなどのLinux distributionを起動できるようになっているのです。

#### ## Cygwinの難点

Linuxのバイナリがネイティブに動かない。かといって、sourceからのbuildも厳しいこと多々。

Cygwin上ですでに動いているawkやsed, sshなどのコマンドを使えただけでも幸せだと思っていた頃もあったのに。

#### ## Virtualboxなどの難点

確かできたような気もしますが、そうはいっても操作を覚えないとWindows-Linux間のファイルのやりとりは大変。

あくまでもx86マシンが提供されるので、Linuxのインストールは別です。

ということは、今に比べるとLinux初心者には優しくはなかったのかもしれません。



## けっこう便利にできている

特定のOSを信じて仰ぐわけではないのですがWindowsというとあのメモ帳が印象にこびりついていて、どうしても親切な設計とは程遠いものを感じていましたが、WSLは違います。

### エクスプローラーからWSLで起動しているファイルシステムを見ることができる！

エクスプローラーに「¥¥wsl\$」をいれるだけで、

なんと今自分の環境を持っているディストリビューションのファイルシステムを見ることができるのです。

## 複数のLinuxを動かすことができる (詳細なやり方は後述)

WSL上のLinuxディストリビューションは、その今の状態を固めてexportができます。

ということはimportができ、同様のものを複数持つことができます。

たとえば今回、Ubuntu-20.04上でhailが動く最小限の環境をつくったとして、それを別の名前にして登録することができます。



### ディストリビューションをexportしimport。別の名前にして登録する。(複数のディストリビューションを持つ)

私の環境では次のようなディレクトリをつくり、複数のディストリビューションを管理しています。

wslから管理される、自分でimportしたディストリビューションを置くディレクトリ C:\Users\ut\wsl

exportしたディストリビューションのイメージファイルを一時的に置くディレクトリ C:\Users\ut\wsl\images

wsl --export Ubuntu-20.04 C:\Users\ut\wsl\images\U2004hail.tar

wsl --import Ubuntu-20.04-hail C:\Users\ut\wsl\Ubuntu-20.04-hail C:\Users\ut\wsl\images\U2004hail.tar

これで、新しくUbuntu-20.04-hailという名前のディストリビューションが登録されました。

ちょっと複雑ですが、やっていることは、

1行目で、ディストリビューションUbuntu-20.04 (wsl -l で表示される名前) を、U2004hail.tarというファイルにexportしています。

2行目ではそれを、Ubuntu-20.04-hailという名前でimportしています。import先はC:\Users\ut\wsl\Ubuntu-20.04-hailというディレクトリにしました。



## 1. DEMO 実機にインストール

### (参考) WSLについて、おまけ

## 使い慣れると、まるでDockerコンテナのように利用できる

wslコマンドにはいくつかのオプションがありますが、その中にはLinux(のシェル)を起動せずにコマンドのみを実行させるものがあります。

ちょっと長いですが次の例で説明します。

```
wsl -d Ubuntu-20.04-hail -u ut --cd /home/ut -- jupyter-lab --ip=0.0.0.0 --no-browser --NotebookApp.token=""
```

-d Ubuntu-20.04-hail

は、動かしたいディストリビューションの指定です。ここでは自分でつくりた最小限のhail環境をexport/importした、Ubuntu-20.04-hailを指定しています。

利用可能なディストリビューションは

```
wsl --list --all
```

で確認できます。

-u ut

動作させる、Linux上のユーザーを指定しています。

--cd /home/ut

WSL環境では起動時にとんでもないディレクトリにいることがありますよね。

例えば/mnt/c/Users/「ユーザー名」とか。

ここではそれを自分のホームディレクトリとなるように指定しています。

これでjupyterからはきちんと/home/ut以下のファイルが参照できます。

-- jupyter-labから最後まで

実行したいコマンドを指定しています。

-- ls

ならlsが実行されます。試してみてください。

このようにして、wslを使うとwindows環境に複数のLinuxを起動させることができ、  
しかもコマンド一発で目的のプログラムを動かすことができるようになります。



## その他、覚えておくと便利なwslのコマンドオプションと簡単な例

### 自分の環境にあるディストリビューションの一覧を取得

wsl --list --verbose

または

wsl -l -v

### オンライン上からインストール可能なディストリビューションの一覧を取得

wsl --list --online

または

wsl -l -o

### オンラインより、ディストリビューションを取得してインストール

wsl --install -d Ubuntu-18.04

これでUbuntu-18.04が入ります。

(Microsoft storeでsearchしてポチポチ操作が不要に！)



### 不要になったディストリビューションを削除

wsl --unregister Ubuntu-16.04

リマインドなしで消されるのでちょっと注意。

### 特定のディストリビューションを停止

wsl --unregister Ubuntu-16.04

### 特定のディストリビューションを起動

wsl -d Ubuntu-16.04



(参考) WSLについて、おまけ

1. DEMO 実機にインストール

