



# Jupyter Notebook が変える、 あなたのチームの運用作業

2018年7月29日 11:15-12:00

国立情報学研究所 クラウド基盤研究開発センター  
特任研究員 長久 勝





# 自己紹介

長久 勝 (Masaru Nagaku)

国立情報学研究所 (NII)  
クラウド基盤研究開発センター  
特任研究員



1994年から2010年まで、ゲームや映像配信などコンテンツ業界を中心に、技術職として、ソフトウェア開発(クライアント・サーバ)、システム構築・運用に従事。2010年から2015年まで、NIIにて、特任技術専門員として、トップエスイー(社会人向け教育プログラム)運営、所内クラウド基盤構築・運用に従事。2015年から2017年まで、オンラインゲームサーバソリューションの企画・開発・運用に従事。現職では、研究・教育分野でのクラウド利用促進の活動に従事。本講演のテーマで大学ICT推進協議会2017年度年次大会優秀ポスター賞。早稲田大学GEC非常勤講師。NPO法人IGDA日本SIG-4NG正世話人。

<https://researchmap.jp/mnagaku>





# 本講演の対象者

- システム運用業務で  
コンソールからコマンドを叩いている方
  
- また、そうした現場で  
後輩をどうやって育てたら良いか迷っている方





# 本講演の対象者

- システム運用業務で  
コンソールからコマンドを叩いている方
  - 手順書と、作業時のメモと、作業証跡保存が、  
ワンストップでイイ感じになる方法を話します
- また、そうした現場で  
後輩をどうやって育てたら良いか迷っている方
  - 先輩のやった作業を、後輩がなぞれるので、  
追体験による技能移転が可能になります

**持って帰れる知見はコチラ**



# 概要

ある日、先輩から、あなたに、「このサーバの調査したいから、この本のここに載ってる一連の手順で、どんな感じか確認しといて。あ、あと、ポートが1つ LISTEN になってて http サーバになってるはずなんで、それが動いてるかも確認しといて。」と指示があったとします。あなたは、ターミナルで、手順に沿ってコマンドを打ち込み、出力を Excel にコピーして、時には、知らないコマンドをググったり、入っていないコマンドをインストールしながら、作業を進めるでしょう。数週間後、同様の調査を指示された時、あなたは前回の成果物と記憶を頼りに、概ね同じ手順を繰り返すことになるでしょう。NII クラウド運用チームが実践する、**Jupyter Notebook** を用いてシステムの構築・運用を行う手法「**Literate Computing for Reproducible Infrastructure**」で、この一般的なシステム運用作業はどう変わるのでしょうか？詳しくお見せします。我々のチームが積み上げてきた経験と知見を共有して、みなさんの毎日の作業が少しでも改善できればと思います。





# 概要

ある日、先輩から、あなたに、「このサーバの調査したいから、この本のここに載ってる一連の手順を、どうか感じを確認して、また、ポッドキャストのLIVENに載せて

**今日は、ムズカシイ話はやめて、  
具体的に、  
「こんなことができます！」  
を見せます**

駿





# デモシナリオ (従来通りコマンド叩く)

M先輩から

「3dwebmapのサーバ、中で何動いてるかレポート作って」

と指示があった

1. ターミナルのログを残す設定にする
2. `sudo docker exec -it jtf2018_3dwebmap_1 bash`  
sshの代わり
3. 「linux os 確認」でググる、`/etc/issue`を見つける
4. `cat /etc/issue`  
OSがdebian8と確認
5. 「debian サービス 一覧」でググる、`insserv`を見つける
6. `insserv -s`  
サービス一覧を表示、それっぽいサービスは見つからず
7. `ps aux`  
プロセスを確認、`node.js`が動いているのを見つける
8. 「listen debian」でググる、`ss`コマンドを見つける
9. `ss -ltnat ; ss -ltnau`  
8000番が開いているのを確認する
10. ブラウザで8000番を確認するとwebアプリが動いているのが分かる

```

+
「linux os 確認」でググって見つけた、/etc/issueを確認する方法を使う
https://eng-entrance.com/linux-os-version

「debian サービス 一覧」でググって見つけた、insservを使う
https://www.mk-mode.com/octopress/2015/06/03/debian-8-service-management/

「listen debian」でググって見つけた、ssを使う
http://mzgwks.com/post/linux-port-confirm/

```

```

jovyan@620718c4ad691:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS
620718c4ad691     magaku/jcr-minimal-notebook   "linux -- bash -c _"    About an hour ago   Up Ab
9cd6306c7b2f     0.0.0.0:8080->8080/tcp         jtf2018_3dwebmap_1     "node server.js --pu"  About an hour ago   Up Ab
40645016040     0.0.0.0:9000->9000/tcp         jtf2018_3dwebmap_1     "python3 /home_     About an hour ago   Up Ab
60645016040     0.0.0.0:5000->5000/tcp         jtf2018_wiki_1        "/bin/sh -c '._/home_     About an hour ago   Up Ab
66682f6cb29d     kyootan/boinc-runtar         "fstartup.sh"         About an hour ago   Up Ab
jovyan@620718c4ad691:~$ docker exec -it jtf2018_3dwebmap_1 bash
Get-Permissions denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://429aa82f-run2/docker.sock/v1.38/containers/jtf2018_3dwebmap_1/json: dial unix /var/run/docker.sock: connect: permission denied
jovyan@620718c4ad691:~$ sudo docker exec -it jtf2018_3dwebmap_1 bash
node@9cd6306c7b2f:/var/www$ cd /home/node
node@9cd6306c7b2f:~$ ps
node@9cd6306c7b2f:~$ cat /etc/issue
Debian GNU/Linux 8 \n \l
node@9cd6306c7b2f:~$ insserv -s
#00:0:boot
#03:0:6:umountfs
#04:0:6:umountroot
#05:0:6:umountfs
#06:0:6:pendigp
#07:0:6:hwclock.sh
#08:0:6:urandom
#09:0:6:urandom
#10:0:6:urandom
#11:0:6:urandom
#12:0:6:urandom
#13:0:6:urandom
#14:0:6:urandom
#15:0:6:urandom
#16:0:6:urandom
#17:0:6:urandom
#18:0:6:urandom
#19:0:6:urandom
#20:0:6:urandom
#21:0:6:urandom
#22:0:6:urandom
#23:0:6:urandom
#24:0:6:urandom
#25:0:6:urandom
#26:0:6:urandom
#27:0:6:urandom
#28:0:6:urandom
#29:0:6:urandom
#30:0:6:urandom
#31:0:6:urandom
#32:0:6:urandom
#33:0:6:urandom
#34:0:6:urandom
#35:0:6:urandom
#36:0:6:urandom
#37:0:6:urandom
#38:0:6:urandom
#39:0:6:urandom
#40:0:6:urandom
#41:0:6:urandom
#42:0:6:urandom
#43:0:6:urandom
#44:0:6:urandom
#45:0:6:urandom
#46:0:6:urandom
#47:0:6:urandom
#48:0:6:urandom
#49:0:6:urandom
#50:0:6:urandom
#51:0:6:urandom
#52:0:6:urandom
#53:0:6:urandom
#54:0:6:urandom
#55:0:6:urandom
#56:0:6:urandom
#57:0:6:urandom
#58:0:6:urandom
#59:0:6:urandom
#60:0:6:urandom
#61:0:6:urandom
#62:0:6:urandom
#63:0:6:urandom
#64:0:6:urandom
#65:0:6:urandom
#66:0:6:urandom
#67:0:6:urandom
#68:0:6:urandom
#69:0:6:urandom
#70:0:6:urandom
#71:0:6:urandom
#72:0:6:urandom
#73:0:6:urandom
#74:0:6:urandom
#75:0:6:urandom
#76:0:6:urandom
#77:0:6:urandom
#78:0:6:urandom
#79:0:6:urandom
#80:0:6:urandom
#81:0:6:urandom
#82:0:6:urandom
#83:0:6:urandom
#84:0:6:urandom
#85:0:6:urandom
#86:0:6:urandom
#87:0:6:urandom
#88:0:6:urandom
#89:0:6:urandom
#90:0:6:urandom
#91:0:6:urandom
#92:0:6:urandom
#93:0:6:urandom
#94:0:6:urandom
#95:0:6:urandom
#96:0:6:urandom
#97:0:6:urandom
#98:0:6:urandom
#99:0:6:urandom
#100:0:6:urandom

```





# デモシナリオ (従来通りコマンド叩く)

M先輩から

「3dwebmapのサーバ、中で何動いてるかレポート作って」

と指示があった

+

「linux os 確認」でググって見つけた、/etc/issueを確認する方法を使う

<https://eng-entrance.com/linux-os-version>

「debian サービス 一覧」でググって見つけた、insservを使う

<https://www.mk-mode.com/octopress/2015/06/03/debian-8-service-management/>

「listen debian」でググって見つけた、ssを使う

<http://mzggkworks.com/post/linux-port-confirm/>

```
jovyan@620718c4ad691:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS
620718c4ad691      mnapaku/jcriminal-notebook   "linux -- bash -c _"    About an hour ago   Up Ab
3cd6306c7b2f      tuangje/3dcitydb-web-map     "node server.js --pu..." About an hour ago   Up Ab
4e145016040       0.0.0.0:8000->8000/tcp        jtf2018_3dwebmap_1     "/bin/sh -c ' /home..." About an hour ago   Up Ab
ut an hour       0.0.0.0:5500->5500/tcp        jtf2018_wiki_1        "/bin/sh -c ' /home..." About an hour ago   Up Ab
6662f6b2ab       kyootan/boinc-zumtec        "Tartartup.sh"        About an hour ago   Up Ab
ut an hour       jtf2018_boinc_1
jovyan@620718c4ad691:~$ docker exec -it jtf2018_3dwebmap_1 bash
Get permissions denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://42Pvaas2#run32#docker.sock/v1.38/containers/jtf2018_3dwebmap_1/json: dial unix /var/z
na/docker.sock: connect: permission denied
jovyan@620718c4ad691:~$ sudo docker exec -it jtf2018_3dwebmap_1 bash
node@9cd6306c7b2f:/var/www$ cd /home/node
node@9cd6306c7b2f:/var/www$ ps
node@9cd6306c7b2f:/var/www$ cat /etc/passwd
root:x:/usr/sbin:/usr/sbin/nologin
node:x:/var/www:/bin/bash
node@9cd6306c7b2f:/var/www$ insserv -s
#03:0 6:umountfs
#104:0 6:umountroot
#102:0 6:umountfsrsh
#101:0 6:memdisge
#103:0 6:hwclock.sh
#101:0 6:urandom
#105:6:zabbix
#102:5:udev
#104:8:hwclock.sh
#103:8:mountdevubfs.sh
#105:8:checkroot.sh
#10:5:urandom
#108:8:mountall.sh
#109:8:mountall-bootclean.sh
```

1. ターミナルのログを残す設定にする
2. `sudo docker exec -it jtf2018_3dwebmap_1 bash`  
sshの代わり
3. 「linux os 確認」でググる、/etc/issueを見つける
4. `cat /etc/issue`  
OSがdebian8と確認
5. 「debian サービス 一覧」でググる、insservを見つける
6. `insserv -s`  
サービス一覧を表示、それっぽいサービスは見つからず
7. `ps aux`  
プロセスを
8. 「listen de
9. `ss -ltnat ; s`  
8000番が開いているのを
10. ブラウザで8000番を確認するとwebアプリが動いているのが分かる

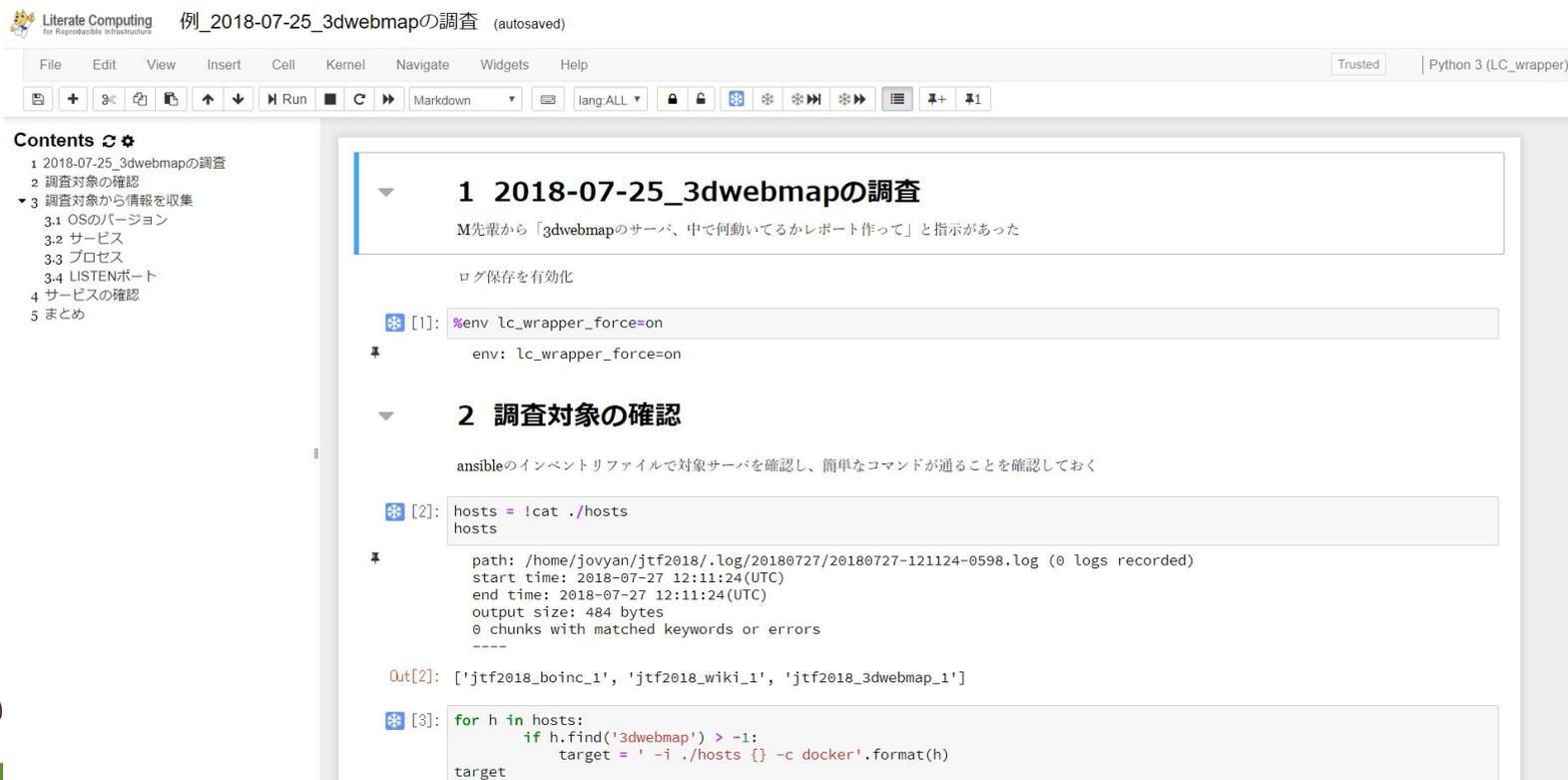
いつも、こんな感じですよ？



# デモシナリオ ( LC4RI) その1

M先輩から

「3dwebmapのサーバ、中で何動いてるかレポート作って」  
と指示があった



Literate Computing  
for Reproducible Infrastructure

例\_2018-07-25\_3dwebmapの調査 (autosaved)

File Edit View Insert Cell Kernel Navigate Widgets Help Trusted Python 3 (LC\_wrapper)

Contents

- 2018-07-25\_3dwebmapの調査
- 調査対象の確認
- 調査対象から情報を収集
  - 3.1 OSのバージョン
  - 3.2 サービス
  - 3.3 プロセス
  - 3.4 LISTENポート
- サービスの確認
- まとめ

## 1 2018-07-25\_3dwebmapの調査

M先輩から「3dwebmapのサーバ、中で何動いてるかレポート作って」と指示があった

ログ保存を有効化

```
[1]: %env lc_wrapper_force=on
env: lc_wrapper_force=on
```

## 2 調査対象の確認

ansibleのインベントリファイルで対象サーバを確認し、簡単なコマンドが通ることを確認しておく

```
[2]: hosts = !cat ./hosts
hosts

path: /home/jovyan/jtf2018/.log/20180727/20180727-121124-0598.log (0 logs recorded)
start time: 2018-07-27 12:11:24(UTC)
end time: 2018-07-27 12:11:24(UTC)
output size: 484 bytes
0 chunks with matched keywords or errors
----

Out[2]: ['jtf2018_boinc_1', 'jtf2018_wiki_1', 'jtf2018_3dwebmap_1']
```

```
[3]: for h in hosts:
    if h.find('3dwebmap') > -1:
        target = '-i ./hosts {} -c docker'.format(h)
target
```

# デモシナリオ (LC4RI) その1

M先輩から

「3dwebmapのサーバ、中で何動いてるかレポート作って」  
と指示があった



**Jupyter Notebook を用いた LC4RI で、  
手順書と、作業時のメモと、作業証跡保存が、  
ワンストップでいい感じになった**

```
path: /home/jovyan/jtf2018/.log/20180727/20180727-121124-0598.log (0 logs recorded)
start time: 2018-07-27 12:11:24(UTC)
end time: 2018-07-27 12:11:24(UTC)
output size: 484 bytes
0 chunks with matched keywords or errors
----
```

```
Out[2]: ['jtf2018_boinc_1', 'jtf2018_wiki_1', 'jtf2018_3dwebmap_1']
```

```
In [3]: for h in hosts:
        if h.find('3dwebmap') > -1:
            target = '-i ./hosts {} -c docker'.format(h)
        target
```





# デモシナリオ (LC4RI) その2

M先輩から

「wikiのサーバ、中で何動いてるかレポート作って」  
と指示があった

「2018-07-25\_3dwebmapの調査」とほぼ同じ作業が予想された  
ので、後輩に任せることにした

Literate Computing for Reproducible Infrastructure 例\_2018-07-26\_wikiの調査 (unsaved changes)

File Edit View Insert Cell Kernel Navigate Widgets Help Not Trusted Python 3 (LC\_wrapper)

Contents ✪

- 1 2018-07-26\_wikiの調査
- 2 調査対象の確認
- 3 調査対象から情報を収集
  - 3.1 OSのバージョン
  - 3.2 サービス
  - 3.3 プロセス
  - 3.4 LISTENポート
  - 3.5 サービスの確認
- 5 まとめ

## 1 2018-07-26\_wikiの調査

M先輩から「wikiのサーバ、中で何動いてるかレポート作って」と指示があった

ログ保存を有効化

```
[1]: %env lc_wrapper_force=on
env: lc_wrapper_force=on
```

## 2 調査対象の確認

ansibleのインベントリファイルで対象サーバを確認し、簡単なコマンドが通ることを確認しておく

```
[2]: hosts = icat ./hosts
hosts
path: /home/jovyan/jtf2018/.log/20180727/20180727-122745-0729.log (1 logs recorded)
start time: 2018-07-27 12:27:45(UTC)
end time: 2018-07-27 12:27:45(UTC)
output size: 555 bytes
0 chunks with matched keywords or errors
----
Out[2]: ['jtf2018_boinc_1', 'jtf2018_wiki_1', 'jtf2018_3dwebmap_1']
```

```
[3]: for h in hosts:
    if h.find('wiki') > -1:
        target = '-i ./hosts {} -c docker'.format(h)
        target
```



# デモシナリオ (LC4RI) その2

M先輩から

「wikiのサーバ、中で何動いてるかレポート作って」  
と指示があった

「2018-07-25\_3dwebmapの調査」とほぼ同じ作業が予想された  
ので 後輩に任せることにした

**Jupyter Notebook を用いた LC4RI で、  
先輩のやった作業を、後輩がなぞれるので、  
追体験による技能移転が可能になった**

## 調査対象の確認

ansibleのインベントリファイルで対象サーバを確認し、簡単なコマンドが通ることを確認しておく

```
[2]: hosts = icat ./hosts
hosts

* path: /home/joyvan/jtf2018/.log/20180727/20180727-122745-0729.log (1 logs recorded)
  start time: 2018-07-27 12:27:45(UTC)
  end time: 2018-07-27 12:27:45(UTC)
  output size: 555 bytes
  0 chunks with matched keywords or errors
  ----
Out[2]: ['jtf2018_boinc_1', 'jtf2018_wiki_1', 'jtf2018_3dwebmap_1']

[3]: for h in hosts:
      if h.find('wiki') > -1:
          target = '-i ./hosts {} -c docker'.format(h)
      target
```



Lite  
for R

# Notebookをこう使う

- 確定的な手順が存在する
  - Notebookが実行できる手順書として作成できる
  - スケジューラからNotebookを叩けば自動化できる
- 探索的な操作
  - 操作履歴や周辺情報などをまとめて残せる
  - 他の人が同様の作業を行うための雛型になる
  - 洗練すると手順書にできる場合がある

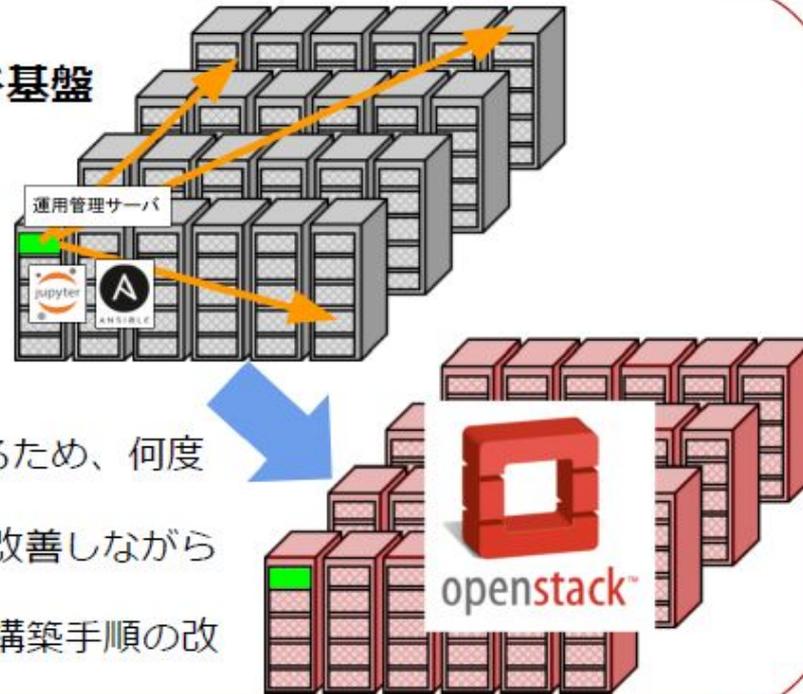
くわしくは、Facebookで！



# NIIクラウド運用チームの実践

## 再構築可能な 研究用ベアメタルクラウド基盤

ネットワークなど最低限の設定が施されたスイッチやサーバの群に対して、運用管理サーバ上のJupyterからNotebookベースの作業で、クラウド基盤を構築できる。手順がNotebook化されているため、何度でも再構築可能である。運用もNotebookベースで、改善しながらの運用が容易である。運用からのフィードバックで構築手順の改善も可能である。



構築後も、現物主義で、Notebookベースの運用を行っている。  
法定停電対応とか、障害調査とか、構成変更とか。





あ、1つ言い忘れてました。  
普通の Jupyter Notebook は  
こんなんじゃないです。



# 構築・運用のために拡張された Jupyter Notebook と、実践している Notebook 記述プラクティス

**セルの実行**

**設計情報、運用状態などの文章**

**実行結果はセル単位でファイルに保存**

**前提条件を満たしていないと失敗する確認用のセル Design By Contract**

**実行結果の表示は要約版 特定の文字列を含むか調べて表示**

**重複実行を防ぐ凍結**

**以前の実行結果 (お手本)**

**再利用率が高く、実行結果が安定している場合、実行結果をお手本に残す**

**成功は緑 失敗は赤 実行中は青でセルの処理状態が分かる**

**コード化された作業内容**

**実行結果をお手本と比較**

**セルのメタデータにmemoを記録 ログファイル内にもmemoを記録 同じmemoのログを集めたディレクトリ**

**Notebook間の関係の図示**

**表示の畳み込み**

**まとめ実行**

**セルの処理状態**

**1 Aironetの設定**

- 1.1 接続を確認する
- 1.2 AnsibleでCssh接続を設定する
- 1.3 fingerprintを記録する
- 1.4 AnsibleでWifiを設定する

**操作対象や再利用の単位で Notebookを分割**

**Notebookの複製で作業する時**

- ・説明が足りていない
- ・例外的な振る舞いに遭遇した
- ・システムの振る舞いが変わっていた場合、追記・修正して、次に備える

**Notebookの複製を渡して作業依頼 スキルトランスファーとして機能**

**パラメタライズできる場合は Notebook冒頭部に パラメータ設定するセクションを付ける**

**直近の類似の作業を行った Notebookを複製して、作業日時を接頭辞に付き、Notebookベースの作業を行う 作業後に残ったNotebookがそのまま証跡となる**

**Notebookをターミナル代わりに使う 紆余曲折があっても全部残す メモの類はmdセルに書く**

**紆余曲折で冗長なNotebookは、適当なタイミングで清書する**

**GUI操作の手順は mdセルに書いて**手作業**する**

**秘密情報の入力など、残ると困るものを扱う場合は、mdセルに書いて、Terminalで**手作業**する**

# 関連情報

- GitHub  
<https://github.com/NII-cloud-operation>  
<https://literate-computing.github.io/index-ja.html>  
<https://github.com/mnagaku/jtf2018> (今日のデモ)
- Facebook  
<https://www.facebook.com/groups/LiterateComputing/>
- お問い合わせ  
[mnagaku@nii.ac.jp](mailto:mnagaku@nii.ac.jp) とか @mnagaku





GakuNin Cloud

# NII

大学共同利用機関法人 情報・システム研究機構

## 国立情報学研究所

National Institute of Informatics



### Literate Computing

for Reproducible Infrastructure