

Literate Computing for Reproducible Infrastructure を用いた WiFi ネットワーク構築手順の作成

長久 勝^{†1} 政谷 好伸^{†1} 谷沢 智史^{†2} 合田 憲人^{†1}

概要: 筆者らは「Literate Computing for Reproducible Infrastructure(以下、LC4RI)」の方法論に基づき、「Jupyter Notebook」を使った、システムの構築と運用を実践している。実践の中で、システム構築や運用に関わるドキュメントの不備や自動化の弊害など、いくつかの既知の問題に対処できることが分かった。また、実践のために「Jupyter Notebook」の機能拡張も行っている。本稿では、LC4RI における構築手順作成の例として WiFi ネットワーク構築の事例を示し、作業個人個人のメモや記憶、暗黙知に依存しない、再構築可能な手順作成手法の実践例を示す。

Construction procedure of WiFi network using Literate Computing for Reproducible Infrastructure

MASARU NAGAKU^{†1} YOSHINOBU MASATANI^{†1} SATOSHI YAZAWA^{†2}
KENTO AIDA^{†1}

1. はじめに

DevOps の概念やプラクティスの浸透により、クラウドネイティブに設計された web アプリケーションを中心として、開発成果物を簡単に繰り返し何度でもデプロイすることが一般的に可能となった。しかし、アプリケーションを載せる基盤の構築においては、簡単に繰り返し何度でも再構築することは、未だに一般的に実現されているとはいえない。

また、システムの構築や運用を、最終的に管理者、即ち人間が責任を持つワークフローであると考えた時、属人性や忘却が問題となる。この問題を、自動化によって解決しようとするアプローチもあるが、効果は限定的である。

筆者らは、これらの問題に対処するための方法論として「Literate Computing for Reproducible Infrastructure (以下、LC4RI)」を提案しており、所内に構築した研究・教育向けクラウド基盤の構築や運用において、実践を行っている[1][2]。

本稿では、LC4RI における構築手順作成の例として WiFi ネットワーク構築の事例を示し、作業個人個人のメモや記憶、暗黙知に依存しない、再構築可能な手順作成手法の実践例を示す。

2. LC4RI の概要

本稿で使う意味での「Literate Computing」は、「IPython」の開発者である Fernando Perez が、2013 年以降、ブログ[3]

や論文[4]で言及しているものを指す。これは、ドキュメントとコードを同じテキストで一元管理する「Literate Programming」[5]の考え方を、IPython のブラウザベースの Notebook スタイルインタフェースである「Jupyter Notebook」(図 1) [6]でコンピュータの操作に適用しようというものである。

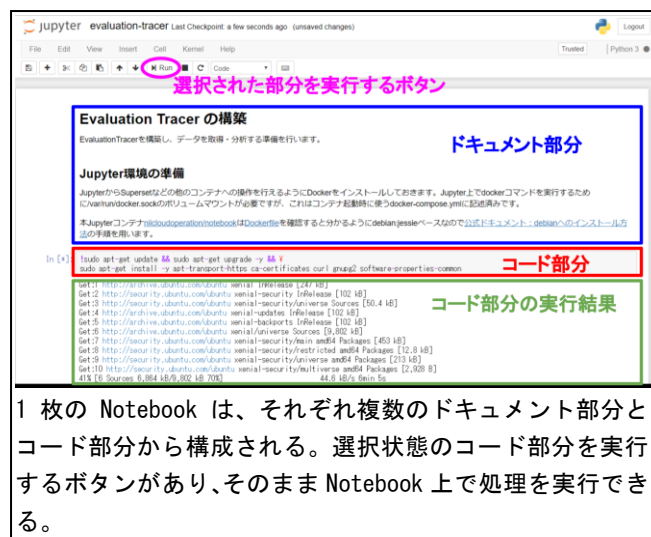


図 1 Jupyter Notebook

筆者らは、この「Literate Computing」を、システムの構築や運用に用いる実践として、Notebook を、そのまま実行可能な手順書として使っている。そのままコードを実行できる UI は、従来必要だった手順書からコンソールへのコピー＆ペーストを不要とするし、実行結果を保存するだけで作業証跡として機能する。

「for Reproducible Infrastructure」とは、構築手順書として正しく動く Notebook があれば、簡単に繰り返し何度でも

^{†1} 国立情報学研究所
National Institute of Informatics
^{†2} 株式会社ボイスリサーチ
Voice Research Inc.

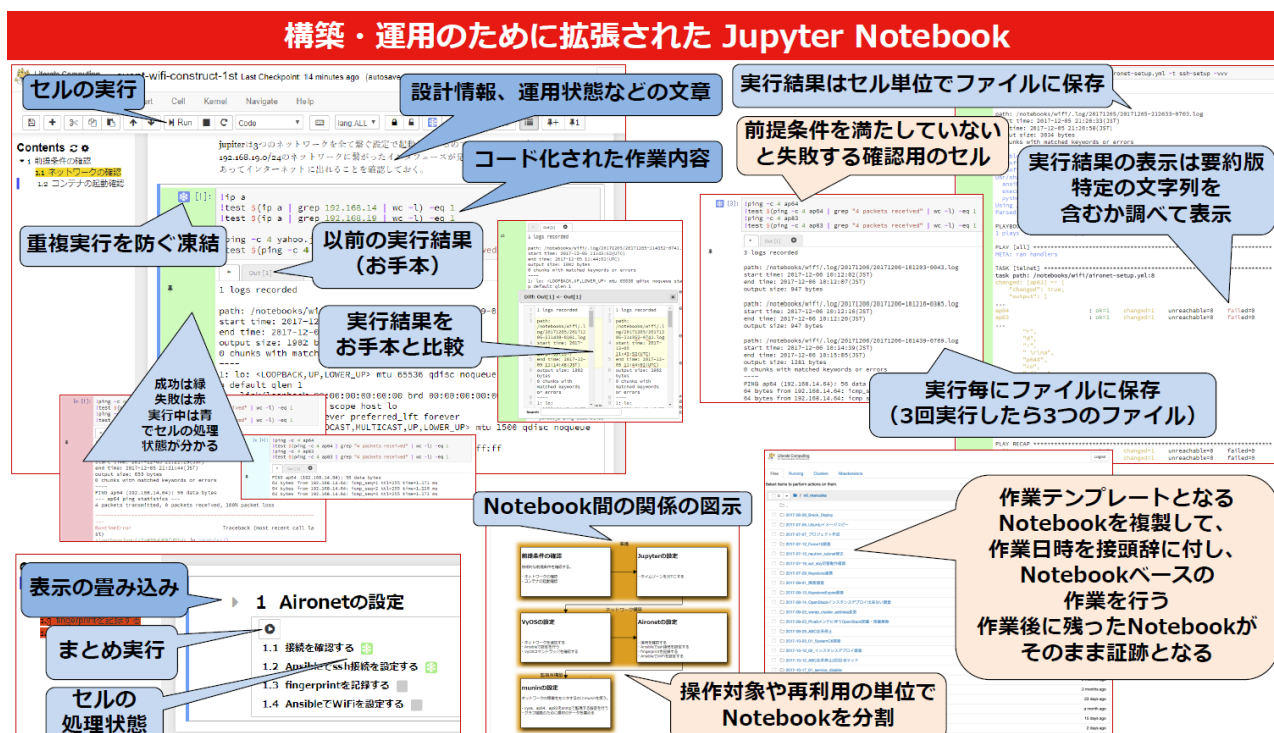


図 2 Jupyter Notebook の拡張について

同じシステムを再構築できるということである。また、ドキュメントとコードだけではなく、実際の実行結果も一体として管理・比較参照可能とすることで、システム固有の事情や、システムの改変の経緯などの情報を留め、運用者間で個々の環境や作業における関心を共有可能とすることで、人間中心の柔軟な自動化を実現するということでもある。

3. Jupyter Notebook の拡張について

Jupyter Notebook をシステムの構築・運用に適用するにあたり、実践の中でさまざまな要望が出たため、プラグインを開発して、それを満たしている。図 2 に描かれた水色の吹き出しの機能は、NII クラウド運用チームで開発したプラグインによって実現されている。

例えば「重複実行を防ぐ凍結」は、システム運用コマンドによくある「2 回実行したらダメ」に対応するため、1 度実行したセルは凍結状態とし、明示的に解凍を行わない限り、凍結セルは実行を指示してもスキップされるようになっている。

以降に述べる Notebook の書き方とシステム構築手順作成の例では、標準的な Jupyter Notebook ではなく、NII クラウド運用チームによって構築・運用業務用に拡張された、図 2 の機能を持つ Jupyter Notebook の使用を前提としていることに留意されたい。

NII クラウド運用チームによる Jupyter Notebook の拡張に関しては、[7]が詳しい。

4. LC4RI の Notebook 記述プラクティス

ここでは、NII クラウド運用チームのメンバが、LC4RI の実践として Notebook を使っている様子を挙げる。先に述べた、Jupyter Notebook の拡張は、構築・運用実務の中で必要を感じて実装されており、ツールとその使い方が互いに影響し合いながら実践が蓄積され、以下に述べる様子にたどりついている。

p1: あるメンバが、ある時に行った一連の作業を 1 枚の Notebook として残す。作業日時と作業内容で命名し、メンバ毎のフォルダに置いておく。実行結果も削除せずに残し作業証跡とする

p2: 新規の作業の場合、Notebook のセルをターミナルとして使う。参考資料などもメモとして併記しながら、作業を進めていく。冗長な作業となっても、まずは全て残す

p3: 以前に行った作業と、途中まで同じ、全体が似ている、全く同じ作業の場合、以前に行った作業の Notebook を複製し(ファイル名は命名規則に沿って新規に付ける)、必要に応じてセルの追加、削除を行いながら、作業を進めていく

p4: 再利用性が高く、実行結果が安定していると見なされる場合、実行結果をお手本に残す

p5: メンバ間で作業依頼を行う場合は、過去の Notebook を渡して情報共有する。ジュニアなメンバの訓練も自然と行える

p6: 同様の作業を何度も行って、種に使えるような Notebook がいくつもある場合、最も直近の作業で使われたものを採用する（操作対象となるシステムは運用の中で変化していくため、過去に使えた Notebook がそのまま使える保証はない。直近の動作実績のある Notebook が最もそのまま使える可能性が高いと考える）

p7: 過去の Notebook を複製して作業する時、抜けている情報に気付いた場合、追記しておく。例えば、システムの一時的な状態（例えばタイムアウトなど）での実行失敗が確認されたら、その可能性を情報として追記しておく。文章での説明と併せて、特定の失敗例として、お手本に追加しておくこともできる（p5 により、今回は、追記された Notebook が使われるため、今回は追記された情報の恩恵を受けられる）

p8: 過去の Notebook に基づいて作業する時、システムの状態が変化していて、記載内容が正しくなくなっていた場合、修正しておく。必要に応じて、お手本も修正する（p5 により、今回は、修正された Notebook が使われるため、今回は Notebook の修正なしに作業できると期待できる）

p9: GUI 操作が作業に含まれる場合は、コード化が難しいので、GUI 操作手順を文章で書いて、カバーする

p10: 秘密情報を入力するなど、実行内容・結果が保存されると困る場合は、セルで実行するべきではないため、操作手順を文章で書いて、Terminal を使って作業する

p11: システムに対し、その作業を行って、意図した結果を得るために、前提があるはず。運用によってシステムの内状態は変化するので、暗黙の前提は危険である。「前提条件を満たしていないと失敗する確認用のセル」を置く。書いた前提条件が足りないことに気付いた場合、その時点で追記を行う（一種の Design By Contract とみなすことができる）

p12: Notebook を清書したくなったら

p12-1: それまでの Notebook 利用で、冗長だと分かっているセルがあったら削る

p12-2: 操作対象や再利用の粒度で、Notebook を分割する

5. WiFi ネットワーク構築手順における実践

<https://github.com/mnagaku/wifi> は、イベントなどで一時的な無線 LAN 接続サービスを提供するため、WiFi ネットワーク構築を行うための手順をまとめたものである。長久が実際に所持する機器の構成に合わせて、ソフトウェアルータ VyOS やアクセスポイント(シスコ社製 Aironet1140)、通信量監視のための Munin の設定を、Jupyter Notebook から行うことができる。

本稿執筆時点で最新の確立した手順として清書された Notebook 群をもつコミットには、「iot40」タグを付与している。master ブランチは、初回コミットから「iot40」タグに至るまで、24 回、コミットが行われており、そのうちの 13 回で、Notebook が作成・更新されている。

長久は、この Notebook 群を書くにあたって、LC4RI の Notebook 記述プラクティスを意識しており、LC4RI を実践する NII クラウド運用チームのメンバと適時議論しながら、作業を行った。

ここでは、Notebook の 13 回の作成・更新のうちのいくつかを取り上げて、LC4RI の Notebook 記述プラクティスがどのように使われたか、また使われなかったかについて述べる。

コミット: f60c25bd5dd6c7570782a3d86cc2bb9bc59295cd [f60c25b]

日時: 2017 年 11 月 19 日 16:11:00

始めて Notebook が 1 枚書かれた。リポジトリ上には、ここまで、個別の動作検証の成果物として

- ネットワーク設定とコンテナ群を上げる docker-compose ファイル
- VyOS 設定用のスクリプトと Playbook
- アクセスポイント設定用の Playbook

が蓄積されてきた。これらを結び付けてシステム構築の手順とするため Notebook が初めて書かれたのが、このコミットである。

p2 より、個別の動作検証も Notebook で行われるべきだったが、作業をカウボーイスタイルで行ったため、個別の動作検証の成果物のみが残っている状態となってしまった。

p2 の実践を怠ったため、

- Aironet1140 において、GUI を使わない設定方法の情報が少なく、かなり時間をかけて、海外の blog 記事から必要な情報を得た。しかしその URL が残っていない
- VyOS に対する Playbook 実行の制約（connect=docker で動かない）を試行錯誤とドキュメント調査で確認したが、そうやって得た技術情報が残っていないとなってしまった。調査で得た情報を残せることを考えると、p2 を実践すべきだったと言える。

書かれた Notebook では、VyOS コンテナにネットワーク割り当てを行い、Ansible の Playbook で設定を行っているが、メモ程度の説明文も含まれておらず、p2 の観点から、出来が悪い。

作業の前提となるネットワークの状態については、Notebook ではなく README.md に書かれているが、Notebook に書かれるべきであった。前提となるネットワークの状態は、要件として具体性があり、p11 の確認用セルが書かれるべきであった。

コミット : 2de3a46bfa12a5d530c092ce82490dcffb7419f3
[2de3a46]

日時: 2017 年 11 月 19 日 17:34:01

VyOS の設定を行う Notebook に、続けて、Aironet1140 に対し、Ansible の Playbook で設定を行う追記を行った。

Aironet1140 に ssh 接続するための設定を行った後、初回の ssh 接続時に fingerprint の確認が求められるが、この振る舞いへの対応に、「kokode ssh yattoku」のメモを書き入れている。この時点では、fingerprint の自動確認の方法が見つけられなかったため、メモの対応となっているが、NII クラウド運用チームのメンバとの議論によって、p10 の観点から、fingerprint の確認は手動で行うべきとの結論を得た。

コミット : be8006af092bf6db0bec764354054a42635a0eda
[be8006a]

日時: 2017 年 11 月 20 日 17:42:28

ここまでの Notebook は、文章による情報の記載がなかったため、情報が不足している状態だった。VyOS と Aironet1140 の設定を行い、WiFi ネットワークを構築する手順の大枠が固まったこともあり、この時点で、文章による詳細な情報の記載が行われた。不足情報を補ったという意味で、p7 の実践と解釈することができる。README.md に書かれた、作業の前提となるネットワークの状態についても、この時点で Notebook に取り込まれている。

コミット : 3d87c490b4e8af6014b5f5d567e7010cc624db3a
[3d87c49]

日時: 2017 年 11 月 20 日 21:13:39

手順の大枠が固まり、文章も補われて、Notebook としての体裁が整ったため、p4 にある、お手本の記録が行われた。以降、Notebook の修正作業に対し、キリのよい段階で、お手本の記録を行うようにしている。

p3 の本来の趣旨から言えば、Notebook の修正作業は、原本に対して行うのではなく、複製し新しい版を作ることで行うべきであるが、今回はその実践が行われておらず、Notebook の変遷をリポジトリから読み解くこととなった。複製し新しい版を作る実践を行っていれば、変遷は、リポ

ジトリに頼らず、Notebook 群から読み解けたはずである。また、複製し新しい版を作る実践を行っていれば、Jupyter-LC_nblineage プラグインの機能による meme の追跡から、変遷を捉えることも可能となっていたはずである。

コミット : 14c9349b83ed7be4ffa9553b8928af2726a56353
[14c9349]

日時: 2017 年 11 月 22 日 13:49:38

通信量監視のための Munin の設定が追記された。この時点で、構築するシステムの構成要素は確定した。

監視には当初、他のソフトウェアを使おうとしていたが、GUI 操作での設定を要求するため、CLI 操作で設定を完結できる Munin を採用した。GUI 操作を要求する作業に関しては p9 での対応が可能だが、構築時の作業時間を短縮するために、p9 が不要となる選択をした。

コミット : af67c1fdcd1ef147bd04013f8d3ccaa9b88269f7
[af67c1f]

日時: 2017 年 11 月 22 日 15:10:49

Notebook の整理を行っているだけで、機能が追加されるような大きな改修は入っていないが、fingerprint についての手作業部分について、見出しの prefix として「[手作業]」を追加し、手作業が存在する旨を強調した。この強調により、畳み込み表示時でも、手作業の存在が一目で分かるようになった。また、作業者の心理状態が、セルを惰性で順次実行していくような状態だったとしても、手作業の存在を明確に伝える/主張する効果を期待できる。繰り返しになるが、これは p10 の実践である。

コミット : de3b818cad4e6959969049bd618dc78e9e57b1bd
[de3b818]

日時: 2017 年 12 月 4 日 18:44:26

作業の前提となるネットワークの状態を確認するセルが追記された。これは P11 の実践である。

この Notebook は、発表などで技術デモとして使われるため、ライブデモでの運用性を考慮し、fingerprint についての手作業部分をコード化した。しかし、先に述べた通り、実運用の場面では、手作業として残すことが望ましい。

コミット : 269adfe98c14067b92f58932ab49f0b0a448797e
[269adfe]

日時: 2017 年 12 月 5 日 15:33:27

セル単位のログファイルを作成し、Jupyter Notebook 上の出力を簡易表示に変更する、Jupyter-LC_wrapper プラグインの機能を使うように、カーネルを変更した。これに合わせて、Playbook 実行時に「-vvv」オプションを付けて、Playbook の実行時出力を詳細化した。

コミット: f9079ec0b4edc0eb6a3076b18351f85c2e51509c
[f9079ec]

日時: 2017 年 12 月 5 日 19:38:03

p12-2 を適用し、Notebook を 5 つに分割し、その関係図も作成した。

6. 実践から得られた知見

本事例では、チームでの運用ではなく、長久が 1 人で構築手順を作成したため、他人とのコミュニケーションのための実践が、積極的に使われる状況にはなかった。このため、チームで共有することに配慮した Notebook ファイルの起こし方に関わる p1、p2、p3、チームでの作業が前提である p5 の実践は使われなかった。p2 が使われなかったため、p12-1 も使われる状況になかった。

また、構築が目的だったため、操作対象の内部状態の変化も考える必要がなかった。このため、p6、p8 の実践は使われなかった。

GUI 操作が手順に入らないようにしたため、p9 の実践も使われなかった。

それら以外の、p4、p7、p10、p11、p12-2 の実践は使われている。これらは、NII クラウド運用チームと全く同じ条件にならないであろう他の現場でも、効果的な適用が可能ではないかと期待できる。

7. おわりに

本稿では、NII クラウド運用チームの実践について、プラクティスを抽出した後、実際に構築手順作成への適用を試み、いくつかのプラクティスの適用状況を確認した。各プラクティスの導入のし易さについて確認できたのではないかと考えている。

今回の事例は構築手順作成であったため、運用に関する知見の適用が部分的なものに限定されてしまった。今後、LC4RI に取り組む他のチームが現れた時、より意味のある比較が可能となり、手法の汎用性を確かめられると考えている。

なお、NII クラウド運用チームで開発した Jupyter Notebook プラグイン群は、OSS として GitHub で公開しているため、誰でも、本稿に挙げた実践を試してみることが可能である。

NII クラウド運用チームの GitHub リポジトリ

<https://github.com/NII-cloud-operation>

また、実際に、クラウド基盤の運用実務で使っている Jupyter Notebook の Docker コンテナイメージも公開している。まずはどのようなものか試してみるのであれば、こち

らを使って頂くこともできる。

Jupyter Notebook server for Literate Computing
(for Reproducible Infrastructure)

<https://hub.docker.com/r/niicloudoperation/notebook/>

実践の中では、まだ解決されていない課題も多く、今後
も取り組みを進めて行く予定である。

参考文献

- [1] 政谷好伸, 谷沢智史, 横山重俊, 吉岡信和, 合田憲人. インフラ・コード化の実践における IPython notebook の適用. 信学技報, 2015, vol. 115, no. 72, p. 27-32.
- [2] Yoshinobu Masatani. Collaboration and automated operation as literate computing for reproducible infrastructure. JupyterCon, 2017.
- [3] Fernando Perez. "Literate computing and computational reproducibility: IPython in the age of data-driven journalism". <http://blog.fperez.org/2013/04/literate-computing-and-computational.html>, (参照 2018-01-31).
- [4] Perez, F. Bussonnier, M. Frederic, J. D. Froehle, B. M. Granger, B. E. Ivanov, P. Kluyver, T. Patterson, E. Ragan-Kelley, B. and Sailer, Z. IPython: components for interactive and parallel computing across disciplines. AGU Fall Meeting Abstracts, 2013.
- [5] Donald Knuth. Literate programming. The Computer Journal, 1984, vol. 27, no. 2, p. 97-111.
- [6] "Project Jupyter". <https://jupyter.org/>, (参照 2018-01-31).
- [7] 長久勝, 政谷好伸, 谷沢智史, 中川晋吾, 合田憲人. Literate Computing for Reproducible Infrastructure による研究・教育環境の構築と運用. 大学 ICT 推進協議会年次大会, 2017.