

Literate Computing for Reproducible Infrastructure

～再構築可能なインフラのための文芸的コンピューティング～

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

2018/01/26

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。

<https://researchmap.jp/mnagaku>

こんなこと、やってませんか？

- ベンダーが構築したシステムに合わせて納品された手順書が間違っていた。そして、それに気づいたのは事故が起こった時だった.....
- pdfの手順書をターミナルにコピーして使ったら、謎の見えない文字が入ってて、動かなかった.....
- 手順書に実行コマンドは書いてあるが、その出力がどうなったら正解か書いていない.....
- 作業証跡がTeraTermのログ。何か起こった時に履歴を遡って調査するのが面倒.....
- リーダがWikiに色々書いてくれても、みんな読まない.....



アジェンダ

LC4RIでは

- こんなことができます
- こんな道具を使います
- こんなやり方をします
- こんな問題が解決できます

こんなことができます

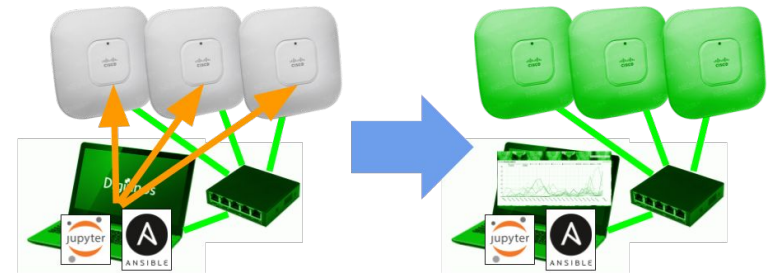


魔改造したJupyterでイベント会場のWiFi環境を爆誕させる話

<https://qiita.com/mnagaku/items/08fd43b99348887cd6>

mnagaku/wifi

<https://github.com/mnagaku/wifi>



本日のデモ

「イベントのためにお借りしたAPなので、簡単に工場出荷状態で返却できるよう、設定を不揮発領域に書き込まない」とゆー、運用に関する前提。構築後に、LAN線の再配線などで、無線APの再設定が必要になった場合のオペレーションを実演する。

こんな道具を使います

Jupyter Notebook

- そのままコードを実行できるUI
(ターミナルにコピー不要)
- マークダウンで文章も書ける
(Wiki同様の情報共有ツールとして機能)
- 実行結果が保存できる
(作業証跡として機能)



⇒実行できる手順書として機能する

こんな道具を使います

構築・運用のために拡張された Jupyter Notebook

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

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)

実行結果をお手本と比較

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

実行毎にファイルに保存 (3回実行したら3つのファイル)

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SSH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

作業テンプレートとなる Notebookを複製して、作業日時を接頭辞に付し、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

前準備条件の確認

Jupyterの設定

YVOSの設定

Aironetの設定

muminの設定

こんな道具を使います

NIIクラウド運用チームによる拡張(OSS)

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

NIIクラウド運用チームがクラウド基盤の構築・運用
実務に使っているJupyterのDockerイメージ

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

今すぐ使える！



こんなやり方をします(今のところ)

構築・運用のために拡張された Jupyter Notebook

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

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

コード化された作業内容

重複実行を防ぐ凍結

以前の実行結果 (お手本)

実行結果をお手本と比較

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

実行毎にファイルに保存 (3回実行したら3つのファイル)

表示の畳み込み

まとめ実行

セルの処理状態

Notebook間の関係の図示

作業テンプレートとなる Notebookを複製して、作業日時を接頭辞に付し、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

操作対象や再利用の単位で Notebookを分割

1 Aironetの設定

1.1 接続を確認する

1.2 AnsibleでSSH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

前提条件の確認

ネットワーク環境を確認する。
- ネットワークの接続
- コンテナの起動確認

Jupyterの設定

タイムゾーンを設定する

vyosの設定

ネットワークの構成を定義する。
- vyosのインストール
- vyosの起動確認

Aironetの設定

ネットワークの構成を定義する。
- Aironetのインストール
- Aironetの起動確認

minininの設定

ネットワークの構成を定義する。
- minininのインストール
- minininの起動確認

こんなやり方をします(今のところ)

- 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
- 新規の作業の場合、Notebook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
- 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で新規に付ける)、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。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こともできる



こんなやり方をします(今のとこ)

- 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
- 同様の作業を何度も行っていて、種に使いそうな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
- 過去のNotebookに基づいて作業する時、抜けている情報に気付いたら追記しておく
- 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ら、修正しておく。必要に応じて、お手本も修正する
- 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

こんなやり方をします(今のところ)

- 過去のNotebookに基づいて作業する時、システムの一時的な状態(例えばタイムアウトなど)での実行失敗があったら、情報を追記しておく。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
- 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部状態は変化するので、暗黙の前提は危険。「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のに気付いたら、その時点で追記を行う
- Notebookを清書したくなったら
 - それまでのNotebook利用で、冗長だと分かっている作業部分があったら削る
 - 操作対象や再利用の粒度で、Notebookを分割する



こんなこと、やってませんか？（解決編）

- ベンダーが構築したシステムに合わせて納品された手順書が間違っていた。そして、それに気づいたのは事故が起こった時だった.....

⇒構築手順がNotebook化されていればシステムの再構築が簡単にできるので、サービスインの前に人為的に事故を起こして手順確認するなども可能

- pdfの手順書をターミナルにコピーして使ったら、謎の見えない文字が入ってて、動かなかった.....

⇒Notebook化されていれば、コピー不要

こんなこと、やってませんか？（解決編）

- 手順書に実行コマンドは書いてあるが、その出力がどうなったら正解か書いていない.....

⇒ 正解例をお手本に入れておける

- 作業証跡がTeraTermのログ。何か起こった時に履歴を遡って調査するのが面倒.....

⇒ セル単位で履歴を追跡できる

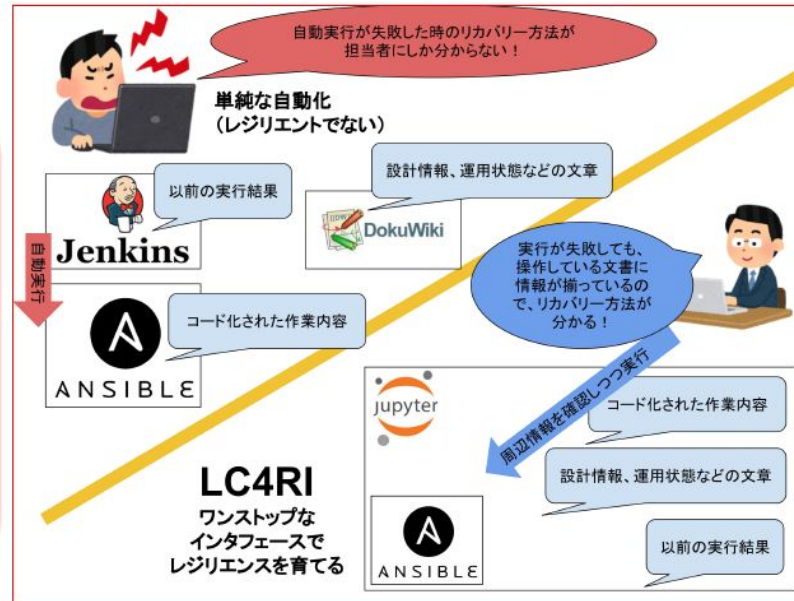
- リーダがWikiに色々書いてくれても、みんな読まない.....

⇒ 作業の単位で、全ての情報がNotebookで一元管理される

こんな問題が解決できます

問題

研究・教育において、限られた計算機リソースを有効に活用するには、必要な期間だけ、必要な大きさの環境を維持するように、構築・運用が必要である。そのためには、**簡単に繰り返し何度でも構築可能**な構築手法が必要である。構築作業の属人化を防いで、作業できる人を増やさないと、簡単に繰り返し何度でも構築可能とは言えない。



解決

自動化技術のサポートを取り入れた構築手法により、**簡単に繰り返し何度でも構築可能**を実現する。それに加えて、**コード化された作業内容、設計情報・運用状態などの文章、以前の実行結果**、を、タスク単位の1つの文書で管理することで、自動化の恩恵を受けつつ、自動化に依存しない、**レジリエントな運用**を行える。

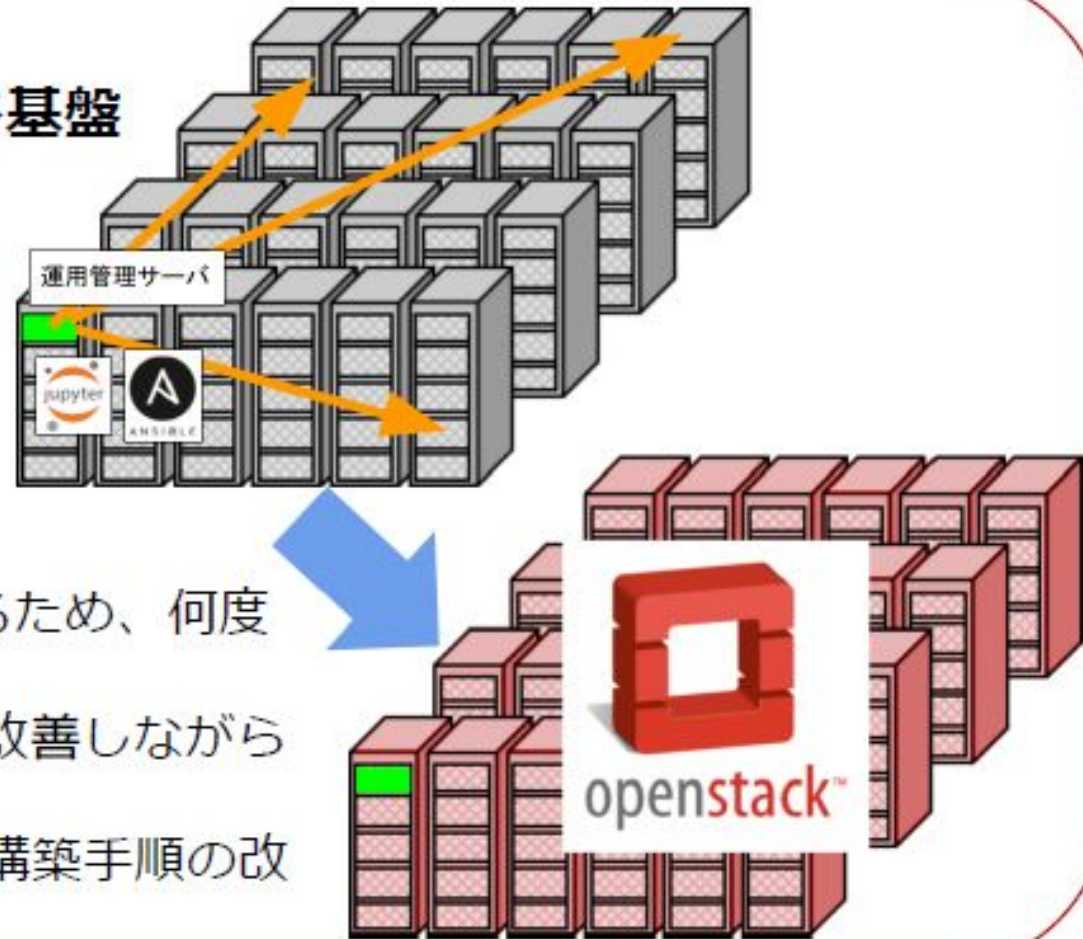
- 「自動化」の先にある問題を解決
 - 「自動化」で属人化がより強まる傾向がある
 - 「自動化」に依存すると、関心が薄れ、リカバリが困難になる
 - 「自動化」範囲は、対象と技術力で決まり、100%にはならない

⇒チームのレジリエンスが育つ実践で乗り越える

こんな問題が解決できます

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Ansibleによる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。



こんな問題が解決できます

オンデマンドクラウド構築サービス

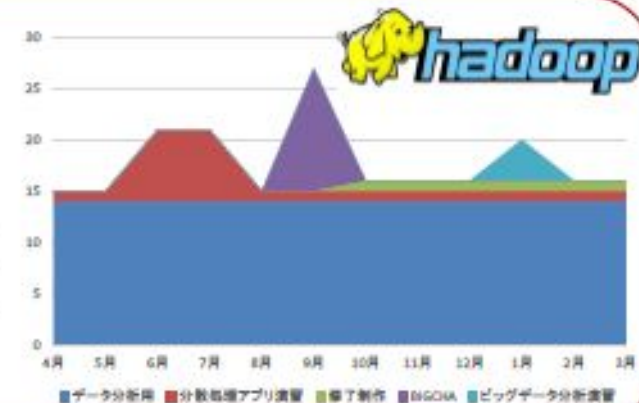
SINET5で接続された商用クラウドに対して、Notebookベースでの環境構築を支援する。

※H30年度中に本格運用開始予定
現在、試験運用中、試用希望者募集中



Hadoop演習環境の構築・運用

Hadoopを利用する複数の講義に対して、それぞれ、必要な期間、必要な規模のクラスタを構築して提供する。途中でクラスタのサイズを変更するケースもある。講師もNotebookベースで操作を行う。



アジェンダ(ふりかえり)

LC4RIでは

- こんなことができます
- こんな道具を使います
- こんなやり方をします
- こんな問題が解決できます

ご清聴、ありがとうございました

LC4RIに関するNIIクラウド運用チームからの情報提供
は下記のFacebookグループから行っています

<https://www.facebook.com/groups/LiterateComputing/>

長久宛てには下記から連絡頂けます

mnagaku@nii.ac.jp

@mnagaku