

Literate Computing for Reproducible Infrastructure

2018年6月14日

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。

<https://researchmap.jp/mnagaku>



質問

- コンソール叩いてLinuxのオペレーションやっています？



Literate Computing for Reproducible Infrastructure (LC4RI)

- NIIクラウド運用チームが2015年から取り組んでいる情報システムの構築・運用手法
- Jupyter NotebookやAnsibleなどを組み合わせて使う

- オンデマンドクラウド構築サービスの操作インターフェースに採用





Contents

- ▼ 1 インシデント調査
 - 1.1 準備
 - 1.2 システムの負荷状態
- ▼ 2 ライブレスポンス実施
 - 2.1 システムの日付と時
 - 2.2 インストールされて
 - 2.3 ファイルシステム情
 - 2.4 OSのバージョン
 - 2.5 カーネルのバージョ
 - 2.6 稼働時間
 - 2.7 cron
 - 2.8 サービス
 - 2.9 ユーザアカウント
 - 2.10 グループ
 - 2.11 ネットワークインタ
 - 2.12 ルーティングテー
 - 2.13 ARPテーブル
 - 2.14 ネットワーク接続
 - 2.15 ロードされているト
 - 2.16 オープン中のファイ
 - 2.17 実行中のプロセスと
 - 2.18 設定情報の保全
 - 2.19 システムログの保全
 - 2.20 ユーザのシェル履
 - 2.21 ファイラー覧
 - 2.22 ライブレスポンス
 - 2.23 システムの負荷状態
- ▼ 3 LISTENポートの確認
 - 3.1 システムの負荷状態
- 4 システムの負荷状態を確
- 5 おしまい

2 ライブレスポンス実施

インシデントレスポンス第3版p179に従って作業を行う。

2.1 システムの日付と時間

```
* [5]: !ansible -a 'date' {target_group}
10.24.128.4 | SUCCESS | rc=0 >>
Mon Jun 18 12:38:37 JST 2018
```

2.2 インストールされているソフトウェア

```
* [8]: !ansible -a 'dpkg --get-selections' {target_group}
10.24.128.4 | SUCCESS | rc=0 >>
accountsservice          install
acl                       install
acpid                    install
adduser                  install
apparmor                 install
appport                  install
appport-symptoms        install
apt                      install
apt-transport-https     install
apt-utils                install
at                       install
base-files               install
base-passwd              install
bash                     install
```

lang: ALL

Icons for file operations: save, copy, paste, undo, redo, run, stop, refresh, etc.

Contents

- 1 インシデント調査
 - 1.1 準備
 - 1.2 システムの負荷状態
- 2 ライブレスポンス実施
 - 2.1 システムの日付と時刻
 - 2.2 インストールされているソフトウェア
 - 2.3 ファイルシステム情報
 - 2.4 OSのバージョン
 - 2.5 カーネルのバージョン
 - 2.6 稼働時間
 - 2.7 cron
 - 2.8 サービス
 - 2.9 ユーザアカウント
 - 2.10 グループ
 - 2.11 ネットワークインターフェース
 - 2.12 ルーティングテーブル
 - 2.13 ARPテーブル
 - 2.14 ネットワーク接続
 - 2.15 ロードされているプロセス
 - 2.16 オープン中のファイル
 - 2.17 実行中のプロセスとPID
 - 2.18 設定情報の保存
 - 2.19 システムログの保存
 - 2.20 ユーザのシェル履歴
 - 2.21 ファイル一覧
 - 2.22 ライブレスポンス実施
 - 2.23 システムの負荷状態
- 3 LISTENポートの確認
 - 3.1 システムの負荷状態
- 4 システムの負荷状態を確認
- 5 おしまい

2 ライブレスポンス実施

インシデントレスポンス第3版p179に従って作業を行う。

2.1 システム

* [5]: `!ansible -a 'date'`

```
10.24.128.4 | SUCCESS | rc=0 >>
Mon Jun 18 12:38:37 JST 2018
```

2.2 インストールされているソフトウェア

* [8]: `!ansible -a 'dpkg --get-selections' {target_group}`

```
10.24.128.4 | SUCCESS | rc=0 >>
accountsservice
acl
acpid
adduser
apparmor
appport
appport-symptoms
apt
apt-transport-https
apt-utils
at
base-files
base-passwd
bash
install
install
install
install
install
install
```

マークダウンで文章が書ける

そのままコードを実行できる

実行結果が保存できる



GUI、CUI、Notebookインタフェース

- GUI(ダッシュボード)
 - とっつきやすい
 - 作り込まれた範囲を越えられない
 - 操作履歴が扱いにくい(スクショ満載になる)
- CUI
 - とっつきにくい
 - 複数のソフトウェアを組み合わせて使える
 - 操作履歴(入力/出力)をログできる
- Notebookインタフェース
 - CUIをGUIっぽく見せる
 - 入出力の他に、文章を書ける(ワンストップ)

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

実行結果をお手本と比較

Notebook間の関係の図示

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

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

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

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**

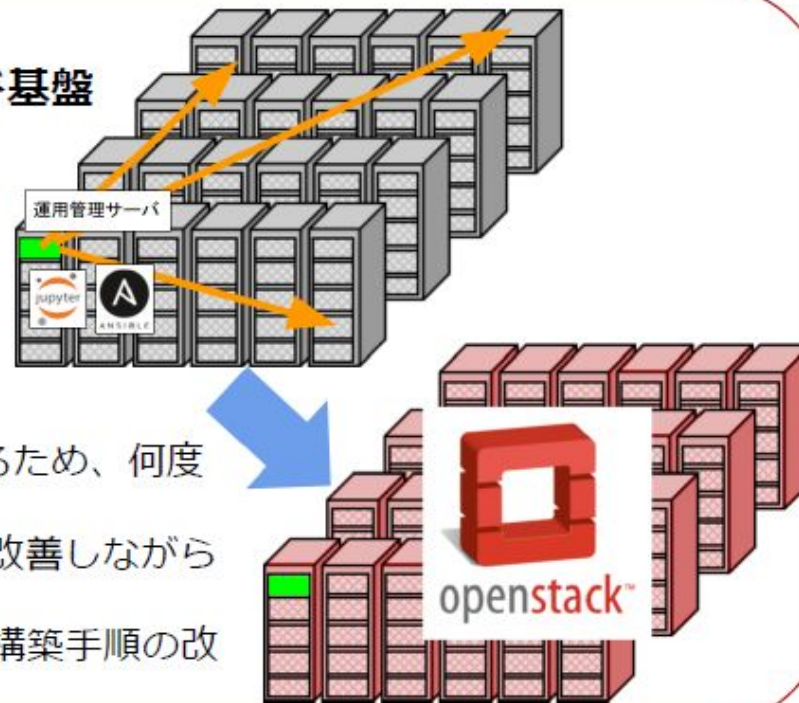
	従来手法	Literate Computing for Reproducible Infrastructure
手順書	紙、pdf	Notebook
作業ツール	teraterm、専用アプリのGUI操作	Notebook (Ansibleなど便利なツールと自由に連携) ※Infrastructure as Codeできない部分を除く。
作業の前提条件	明示されない場合がある 記述が曖昧な場合がある	Notebookの冒頭に、前提条件に関する記述および確認するコード、期待する結果を記載しておく(書き方がガイドされる)
作業の正しさの確認	明示されない場合がある	Notebookに「お手本」として記載しておく (書き方がガイドされる)
証跡	スクショをExcelに貼る teratermのログ	Notebook内の実行出力部に残る
作業の再現性	手順書と対象システムに依存	対象システムに変化がなければ、以前のNotebookが再利用できる
技能移転	チームや個人に依存	十分な情報をNotebookに記載することで、Notebookを媒介に可能
システムの納品	現物	構築のためのNotebook一式 (納品されたNotebook一式で構築できることを確認し検収)
手順書の保守	組織やチームに依存	Notebookが動かなくなった場合、修正して使う (次回は修正したものを再利用する)



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。





オープンフォーラム/オープンハウス

- ハンズオン
 - 本日16:45-18:00 20階2001AB室
 - 若干席に余裕がありますので
参加希望の方は後程お声がけください
- 展示
 - 明日からのオープンハウス [A18]ブース
- お試し環境
 - ハンズオン、展示ブースで、
2週間有効なアカウントを配布
 - ハンズオン、オープンハウスに参加されない方で、
希望される方は後程お声がけください





関連情報

- GitHub

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

<https://literate-computing.github.io/index-ja.html>

- Facebook

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

- お問い合わせ

mnagaku@nii.ac.jp

@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