自動化のススメ！

AnsibleとかOpenTofuとかcloud-initとか

自己紹介

- ぐすくま(@guskma@abyss.fun) アビス丼 鯖缶やってます
- Linux好きのNWエンジニア、Ciscoだいすき
- サーバ系では仕事ではRHEL、趣味はUbuntuとか
- ふぇでば略歴
  - 2017年4月 第一次マストドンブームに乗って JPに登録
  - 2017年9月ごろ？ Ciscoコンソール風Webクライアント「Tooterminal」開発
  - 2017年11月 メイドインアビス テーマ鯖「Abyss.fun」公開
  - 2018年11月「Mastodon meet-up」運営メンバー
  - 2018年？ 分散SNS鯖缶向けDiscord鯖「鯖缶工場」
  - 今年はぼちぼちFediLUGで発表させてもらってます。
- 以降 惰性

今日のテーマ
「インフラ自動化入門」

はじめに

- 前回の発表内容：ゼロトラストについて
  - KADOKAWAが大変なことになりましたね。
  - まさに発表した内容そのままの攻撃手法が使われた感じです。
  - 前回の資料、前々回の資料

- 今回のテーマ：インフラの自動化
  - これが本来の僕の本職。
  - 「自動化ってなんぞや？」という人向けの入門編です。
  - 発表資料も後ほど上げます。

インフラ自動化って何？

インフラ自動化とは

インフラが自動で

構築/運用されることです

インフラ自動化って何？

- 今まで：**手作業**でコマンドを打って**その場で**作業
- 自動化後：**プログラミング的手法**で**事前に準備**
- 関連用語：
  - 構成管理ツール
  - IaC (Infrastructure as Code)
  - プラットフォームエンジニアリング

例えばこういうケース：自動化を始めたきっかけ

- 社会人1年目の現場…
  - 1000拠点以上のネットワーク機器設定変更
    - 全て夜間作業
  - 繰り返される単調な作業
    - コンフィグ投入
    - 状態確認
    - 設定保存
- → 1年半ほどひたすら続く

その時
僕は思いました

→結果

Terratermマクロや

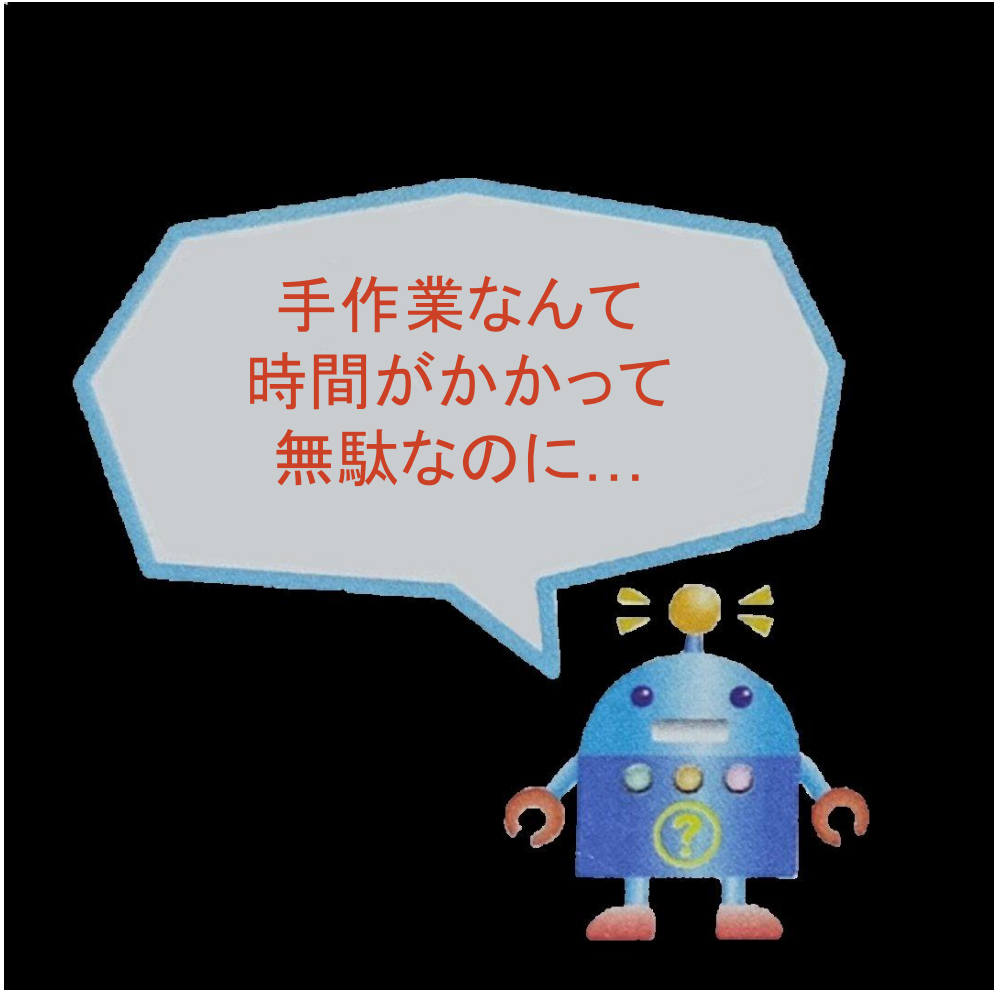
VBAやVBScriptを駆使して、

ある程度オペレーション挟みつつも

半自動化する程度には成功 しました

…それからPerlとかexpectを経て

Ansibleに至ります……



自動化することによるメリット（１）

- 作業時間(工数)の削減
  - 手作業で**１日かかる** 作業
    → **１時間** で終わりました！
  - **確認者が必要** な作業
    → チェックも自動化するので **必要ありません** ！

- 作業品質の向上
  - **コマンド打ち間違え** ました…
    → 当然**打ち間違いはありません** ！
  - **他の作業でしていた** 確認作業をしてない…
    → **全作業共通** の確認作業ができます！

自動化することによるメリット（２）

- システムの標準化
  - システム毎に**構成が違う**…
    → 自動化できる範囲で動くシステムに**統一します**！
  - システム毎に作業**手順が違う**…
    → **全作業共通**の確認作業ができます！

- 作業統制の強化
  - 容量削減したいから**手作業で** rm -rf / したいんだけど…
    → 手作業は**不具合出すからだめ**です！
  - telnetで入りたいから**ポート開けたい**んだけど…
    → セキュリティ意識が甘く**侵入される**からだめです！

自動化する際のデメリット

- 学習コストが高い
  - インフラエンジニアが **専門としていない分野**
    →**プログラミング的手法** 、**Python**に関する知識
  - Gitによる**リポジトリ運用**
    →**Git**の概念、**CLI**コマンド、**GitHubなどのサービス** 利用方法
- 柔軟な対応ができない
  - **あらかじめ決められた対処** しかできない。
  - **ちょっとしたエラーでも切り戻し** にせざるを得ないケースも。
- 小規模システムに対してはあんまり意味がない
  - 単一のシステムで採用しても**意味がない** 。
  - **似た構成や頻繁に作り変える** 構成のほうが効果が目に見える。

自動化を検討すべき人/企業

- 比較的大企業で自社サービスを扱っている
  - 運用コスト削減、ルールの厳格化、不正操作やヒューマンエラーの防止に繋がります。

- 中小企業で顧客のシステム開発/構築受注を行っている
  - 自動化を提供パッケージとして確立しましょう。
  - 人件費抑えて見積もり出せるのでお客様に喜ばれます。

- 検証環境で頻繁に遊ぶ人
  - せめてcloud-initは用意しとくと楽です。
  - あとは忘れやすい作業とか。

自動化する方法を覚えるには

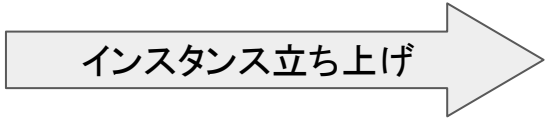
- やっぱり**個人で試せる環境を作る** ことが大事です。

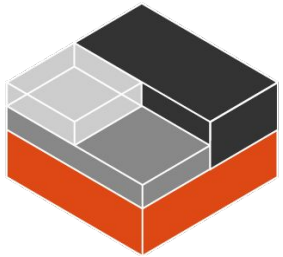
- なので今回は**比較的安価に自動化を試せる環境** を紹介します。

OpenTofu x Ansible x cloud-init を使った Incus仮想化基盤を自動化する手法の紹介

→Incus

OpenTofu x Ansible x cloud-init を使った Incus仮想化基盤を自動化する手法の紹介





仮想化基盤

ここでデモを見せます

……の予定でしたが

ごめんね

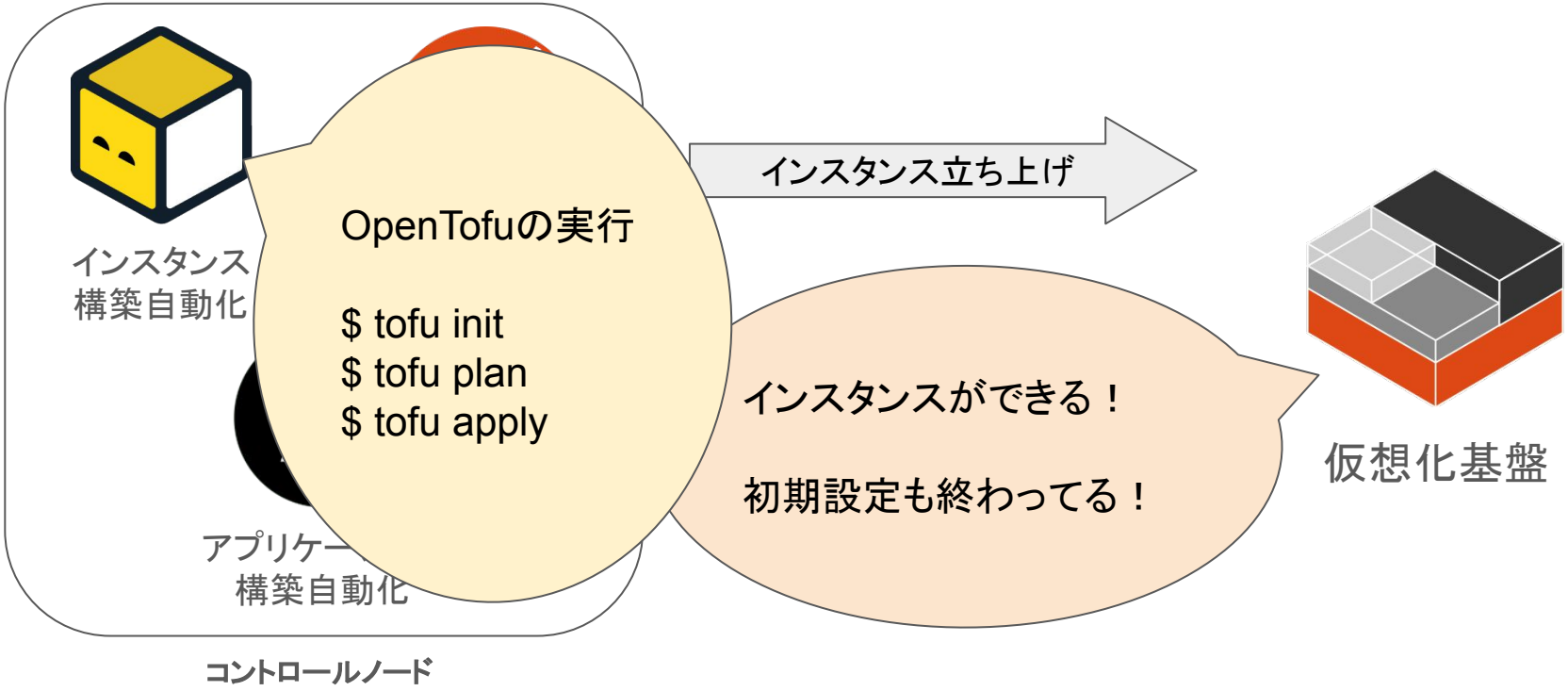
実装の流れと

各ツールの説明だけするね

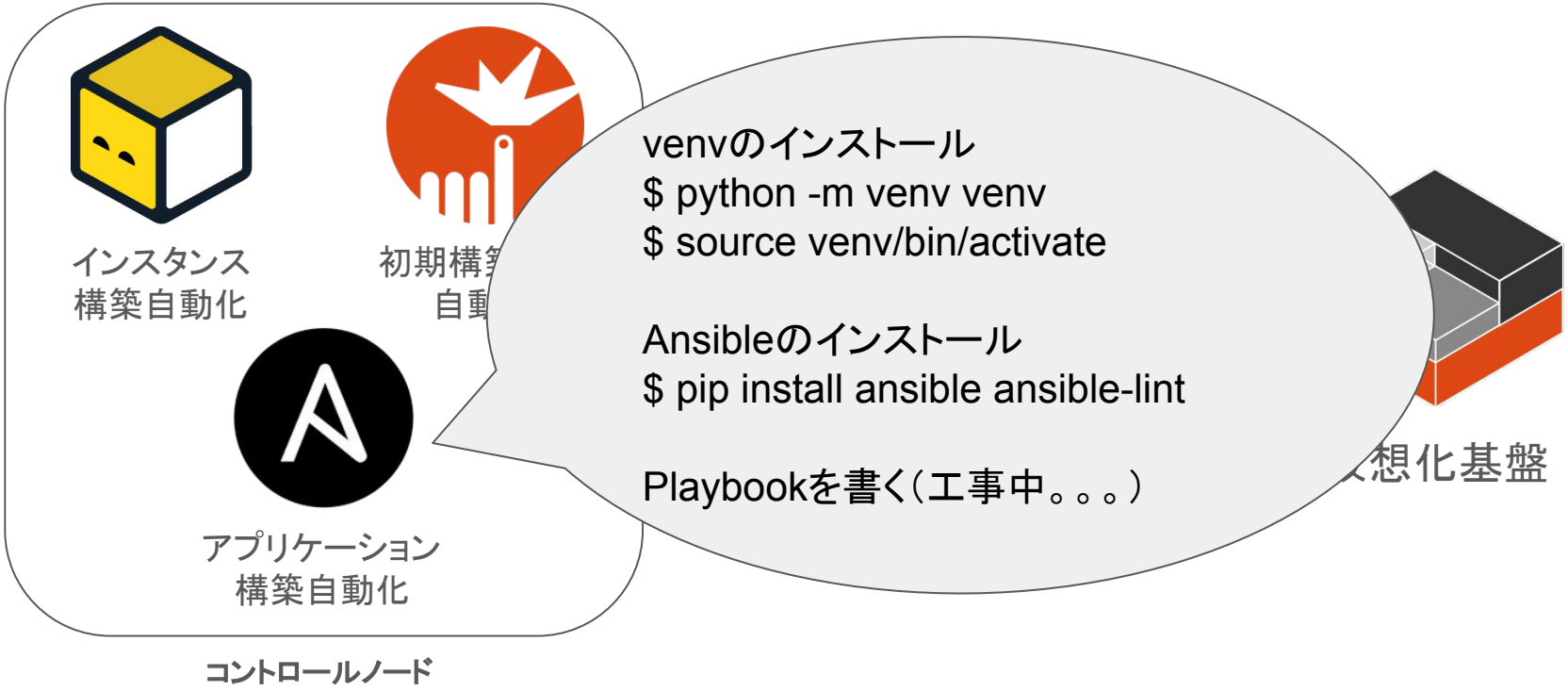
OpenTofu x Ansible x cloud-init を使った Incus仮想化基盤を自動化する手法の紹介（Re:仮）

OpenTofu x Ansible x cloud-init を使った Incus仮想化基盤を自動化する手法の紹介（Re:仮）

OpenTofu x Ansible x cloud-init を使った Incus仮想化基盤を自動化する手法の紹介（Re:仮）

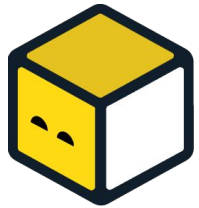


OpenTofu x Ansible x cloud-init を使った Incus仮想化基盤を自動化する手法の紹介（Re:仮）



各プロダクトのざっくりとした解説（1）

- Ansible
  - インフラ自動化としていちばん有名なのがこれかと。
  - 「SSHやWebAPIで操作できる作業」全般をプログラミングする感覚で自動化することができるツール。
  - 幅広いシステムに対応しており、汎用性が高い
  - 自由にできる反面管理が煩雑になりがち。

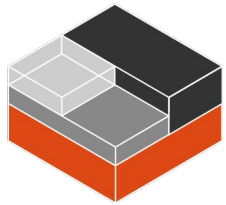


- OpenTofu
  - TerraformのOSS版って言えばわかる人いるかも
  - 汎用性の高いAnsibleに対して仮想化基盤の自動化に特化したツール
  - AWSに詳しい人であればCloudFormation、AzureであればAzureResourceTemplateに相当する。
  - IaaSを中心に幅広い仮想化基盤に対応しており、宣言型でステートフルな仕様のため管理が煩雑になりにくい。

各プロダクトのざっくりとした解説（2）

- cloud-init
  - パッケージのインストール、ユーザの作成など、Linuxの初期設定を自動化するためのツール。
  - AnsibleはSSHが使える必要があるため、cloud-initを使ってそのセットアップを行う

- Incus
  - LXDのOSS版って言えばわかる人いるかも
  - LXDはkvm、xenと並びLinux環境で使える仮想化技術の一つです。
  - 開発プロジェクトの方で去年ゴタゴタがあり、Incusとして分離したようです。
  - まだLXDの色が濃いようですが、徐々に独自機能が増やし別路線を歩むようです。



様々な環境での自動化手法

- サーバ (クラウド/IaaS) の自動化
  - Terraform(OpenTofu)、Ansible、cloud-init
  - 一番自由度が高い
- サーバ (オンプレ/仮想化) の自動化
  - OpenTofu(Terraform)、Ansible、cloud-init
  - OpenTofu(Terraform)が使えるかは仮想化基盤次第
- ネットワーク機器の自動化
  - ほぼAnsible一択

→ 各環境に適した手法があります

その他の自動化手法

- DX (デジタルトランスフォーメーション)
  - アナログな経営からデジタル経営へ
- ビジュアルプログラミング
  - コードを書かずにプログラミング
  - IFTTT、huginn、zapier、n8nなど
- マクロ、スクリプト
  - シェルスクリプト、Teratermマクロ、VBAなど

自動化について少しは理解が深まったでしょうか。

- 自動化の基本概念
- メリットとデメリット
- 様々な手法と適用場面

→ 自動化導入のハードルが少しでも下がれば幸いです

蛇足：今回のスライド作成は少しAIを活用してみました

Claude 3.5 sonnetを使用

やりかた：

1. アイデアを一気にテキスト化
2. AIに入力
3. AIが資料構成を提案
4. 人間が調整・仕上げ

→ 発表資料作成の効率化に貢献

最後に:メイドインアビスの紹介

メイドインアビスは世界の果てにある孤島に存在する大穴「アビス」を探検する少年少女たちを描いた冒険ファンタジーだよ！

Amazonプライムビデオで アニメ1期、劇場版、2期 と配信されているから良かったら見てね！

参考：cloud-init 設定ファイル例

```
#cloud-config
timezone: "Asia/Tokyo"
ssh_pwauth: false
packages:
  - git
  - curl
users:
  - name: ansible
    primary_group: ansible
    lock_passwd: true
    groups:
      - adm
      - sudo
      - lxd
    shell: "/bin/bash"
    sudo: "ALL=(ALL:ALL) NOPASSWD: ALL"
    ssh_authorized_keys:
      - 'SSH公開鍵'
```

参考：OpenTofu 設定ファイル例（1）

```
// プロバイダまわり
terraform {
  required_providers {
    incus = {
      source = "lxc/incus"
      version = "0.1.1"
    }
  }
  backend "local" {
    path = "terraform.tfstate"
  }
}

provider "incus" {
  # Configuration options
  generate_client_certificates = true
  accept_remote_certificate    = true

  remote {
    name     = "ubuntu2204-test"
    scheme   = "https"
    address  = "10.0.0.2"
    default  = true
  }
}
```

```
// プロファイルまわり
resource "incus_profile" "tofu" {
  name = "tofu-profile1"

  device {
    type = "disk"
    name = "root"

    properties = {
      pool = "default"
      path = "/"
    }
  }

  device {
    type = "nic"
    name = "eth0"

    properties = {
      network = "incusbr0"
    }
  }

  config = {
    "cloud-init.user-data" = file("cloud-init.yml")
  }
}
```

参考：OpenTofu 設定ファイル例（2）

```
// インスタンスまわり
resource "incus_instance" "tofu" {
  name  = "tofu-ansible"
  image = "images:ubuntu/noble/cloud"
  profiles  = [incus_profile.tofu.name]
  type = "container"
}

// Ansibleまわり（工事中。。。）
```