

PC 管理省力化のための Ansible の使い方

小畑 沙生* 白石 啓一**

How to use Ansible to save labor for manage PCs

Saki OBATA and Keiichi SHIRAIISHI

Abstract

In today's society, almost all organization have a huge number of PCs. It is extremely inefficient to manage all of these PCs manually. Therefore, it is needed to use a configuration management tool to manage various PC settings and software. We would like to manage 200 PCs in the Information Infrastructure Center's computer room with Ansible configuration management tool. In this paper, 4 virtual PCs installed CentOS are used as an experimental environment. One virtual PC that Ansible is installed on is a controller, the others are targets. The controller manage targets with Ansible. Ansible configuration files that is called "Playbook" are shown. These playbooks are for copying files and install software.

Keywords : Configuration Management Tools, Ansible

1. 緒言

現代社会ではPCで作業を行う企業が数多くあり、各組織で莫大な数のPCを所持している。1000台のPCを手作業で管理すると、1台のPCにかかる時間を2時間とした場合、“ $2\text{h} \times 1000 = 2000\text{h}$ ”より2000時間、1日8時間の作業時間なら250日間の作業時間を要することになる。このことより、PCを手作業で管理するのは非常に非効率である。そこで構成管理ツールを用いてPCの様々な設定やソフトウェア管理等が一括でできると便利だ。

構成管理ツールとは、処理を自動化するという観点ではなく、最終的にマシンを目的の状態にする、という観点で作成されたツールのことである¹⁾。構成管理ツールの導入により、攻めの方向、新規サービスの高速投入などができるようになった。このことから、近年構成管理ツールの需要が高まっている。従来のオンプレミス環境におけるインフラ構築作業では、エンジ

ニアが手動で複雑なオペレーションを実施するのが、当然のこととして捉えられてきた。例えば、ハードウェアのセットアップから、OSのチューニング、またミドルウェアのクラスタ構築に至るまでが手作業であり、一度構築したものを保守期間が切れるまで長期的に利用し続けるという塩漬け状態が、よく見られる運用スタイルであった。また、記録システムでは、企業の基幹系業務やマスターデータを管理する汎用系システムが多く、投資対効果に基づいた業務効率化を図ることが主業務とされている。そのため、ウォーターフォール型の開発プロセスで、信頼性の高いシステムを一度に作り上げることが重視されていた。さらに塩漬け状態のシステムは、運用管理を行う中で、構成管理と定期的に同期を行っていればよく、手作業での構成管理でも、運用工数の計画範囲に十分に収まる状況であった。

ところが、クラウドの登場により、この管理プロセスに大きな影響を与えた。クラウド環境は、仮想化により物理的な制約を排除しただけではなく、APIからの構成情報の取得や操作を可能とした。そのため、ネットワークやストレージボリュームなどのインフラリソー

* 香川高等専門学校専攻科 電子情報通信工学専攻

** 香川高等専門学校 通信ネットワーク工学科

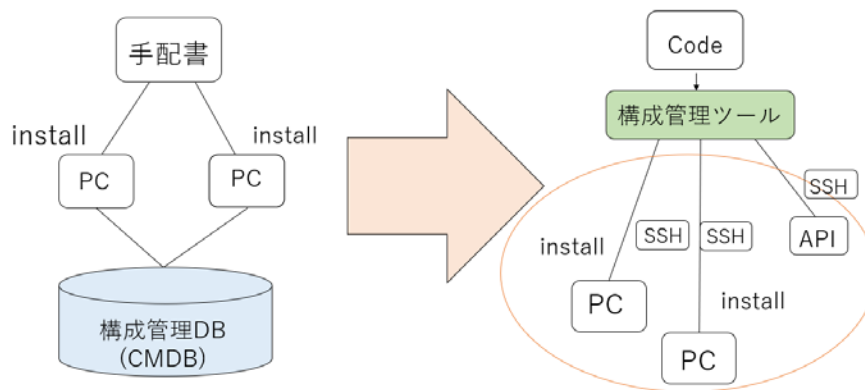


図1 構成管理の変化

スを簡単に把握し、調達、変更、削除できるクラウド環境では、今までのように負担の大きな構成管理が必要なくなった。逆にシステムの構成管理を手動で行っている、インフラリソースのライフサイクルの短さからくる変更頻度の増加により、かえって運用コストを肥大化する可能性がある。

そこで、インフラリソースを効率よく管理するために開発されたのが、システムオペレーションの自動化ツールである。これまでの構成管理の対象は、システムの変更履歴や構成情報そのものであった。しかし、クラウドを活用することで、クラウドプラットフォームに任せることが可能となる。構成管理の役割は、構成情報を取得して意図する状態にシステムを収束することだ。つまり、動作しているシステムの状態を管理することが、構成管理ツールの役割となる。構成管理の変化を図1に示す。このように、クラウドの登場によってオペレーションの自動化、さらには構成管理ツールの実装機能そのものが、構成管理の役割になった。

Chef や Puppet, SaltStack をはじめとする、数多くの構成管理ツールが世の中には存在している。今回は構成管理ツール Ansible を用いて仮想環境 PC の管理サーバから管理対象サーバへ mariadb と PHP をインストールすることを目的とする。Ansible とはアメリカの株式会社 RedHat 社の Python 製の構成管理ツールである。

2. 研究環境

- CentOS 8: Red Hat Enterprise Linux (RHEL) をもとに作成された Linux ディストリビューションである。CentOS-8.1.1911-x86_64 (以下 CentOS 8) を作成した仮想 PC にインストールする²⁾。

- Oracle Virtual Box: 米国オラクル社が開発した仮想化ソフトウェアパッケージである。本研究では仮想 PC を作成するために使用する³⁾。
- Ansible: RedHat 社が開発している構成管理ツールである⁴⁾。システムオペレーションの自動化を支える Python 製の構成管理ツールである。Ansible を利用することによって、スクリプトや手動で行っていたオペレーションを自動化し、運用プロセスを効率化できる。Ansible は構成管理の自動化に伴う、動的な作業タスクの実行制御やマルチスレッド処理など多くの特徴を持っている。主な特徴は、以下の3点が挙げられる。

(1) 設定ファイルの可読性の高さ

Ansible では、処理を記述したファイルは“YAML”というデータ表現記法で記述される。YAML の構文は、インデントと改行に構造的な意味を持つ。それさえ把握できれば特別なコーディングスキルがなくても、どのような処理を行っているのか把握することが可能だ。Chef, Puppet, saltstack などのほかの構成管理ツールでは、プログラム言語に慣れているメンバーばかりで使用するのであれば学習コストも低く迅速に共有できるが、複雑な処理を書くほどプログラミング経験のないエンジニアは、理解に時間を要する。可読性の高い YAML での記述は、エンジニアの手記学習コストを下げるだけでなく、複数で作業を共有できる環境づくりにも貢献する。ただし、Ansible をより効果的に利用するためなら、可動性の高さを重視しながら属人的なコードを排除することを心掛ける必

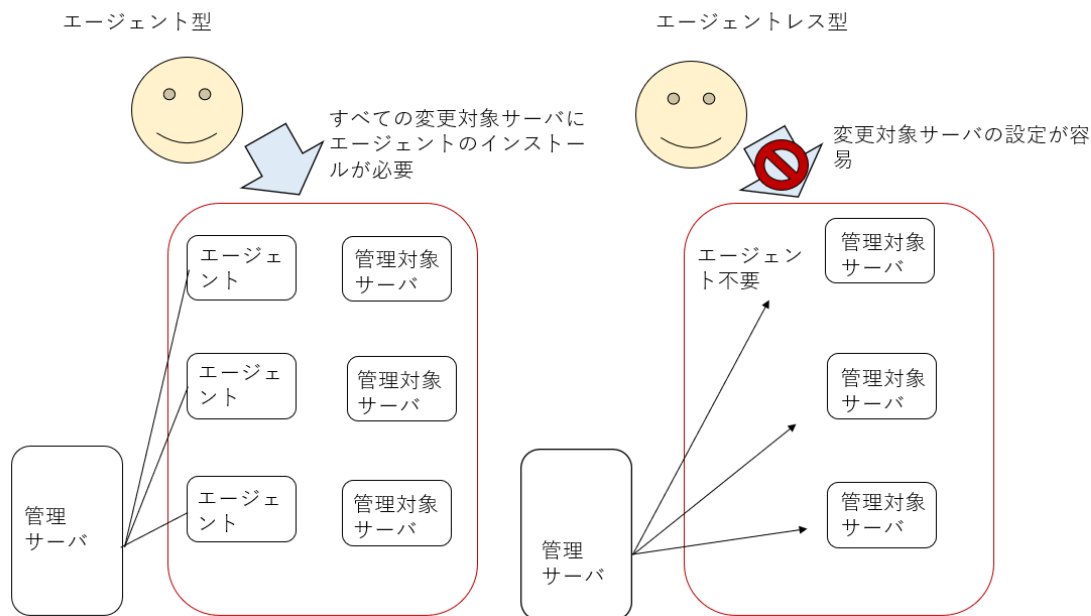


図2 エージェントレス型のアーキテクチャ

要がある。

(2) エージェントレス構成

Ansibleは、構成管理対象サーバに、エージェントソフトウェアをインストールする必要がない。Ansibleが実行する処理内容は、管理サーバからSSH接続を介して安全に送信、実行される。管理サーバや構成管理対象のサーバに特定バージョンのPythonをインストールする必要はあるが、主なLinuxディストリビューションであれば、基本構成としてPythonもSSHサーバもインストールされているので追加の作業が不要である。図2にエージェントレス型とエージェント型の比較を示す。

通常、構築や運用のオペレーションを行う場合、セキュリティ上の観点から作業対象サーバへは、特定の踏み台サーバからのみSSH接続を許可する。既存のオペレーションをAnsibleに移行したとしても、SSH接続を介して、タスクが処理されるため、特別な設定を追加しなくてもよいことがAnsibleの利点である。この特徴が、Ansibleの導入コストを下げる要因にもつながる。

(3) マルチレイヤ対応

Ansibleはマルチレイヤに対応した構成管理ツールである。従来の構成管理ツールでは、パッチの自動やミドルウェアクラスタの動

の設定変更など、ある特定のオペレーションだけを対象にした自動化が主流であった。しかしAnsibleでは、デフォルトの機能だけでも管理対象レイヤが広く、インフラエンジニア、アプリケーションエンジニアの隔てなく利用できるツールとなっている。さらにマルチベンダーの製品もサポートしており、ネットワークベンダーから、エンタープライズOS、クラウドリソースまで、あらゆる製品をコントロールできるのも魅力である。

- MariaDB: リレーショナルデータベース管理システムMySQLをフォークした製品である。最新機能を取り込んだセキュリティや機能を高めている⁵⁾。
- PHP: Webサーバの機能を拡張し、動的にWebページを生成するために用いられるプログラミング言語である。スクリプト言語、軽量言語と呼ばれ、オペレーティングシステム(OS)やアプリケーションソフトの動作や機能などをプログラムの形で記述できる⁶⁾。

3. 研究結果とその検討

3.1 mariadb

(1) 構築環境

VirtualBoxにCentOS8をインストールし、仮想マシンを構築した^{1,7)}。また、Ansibleを用いて仮想マシンに管理サーバ(IP:10.0.2.4)のhostsファイルを管理対象サーバ3台(IP:10.0.2.5, 10.0.2.7, 10.0.2.8)に送

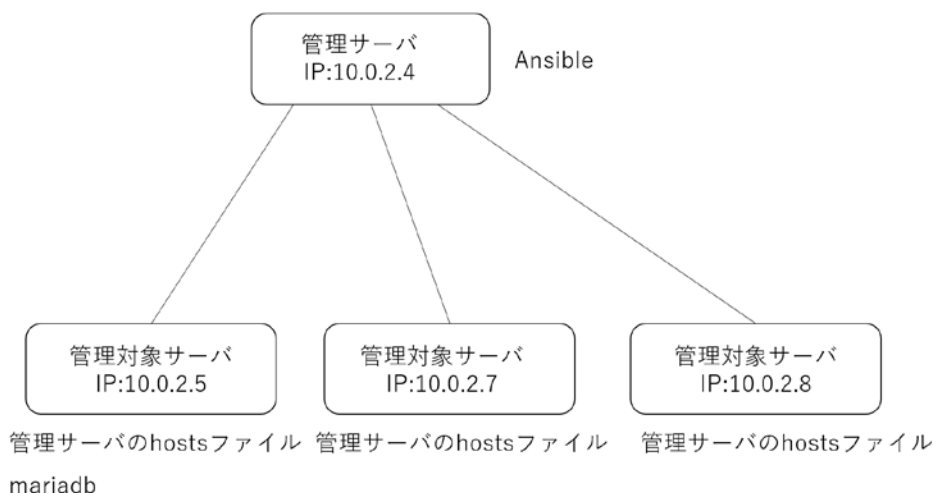


図3 MariaDB 構築環境図

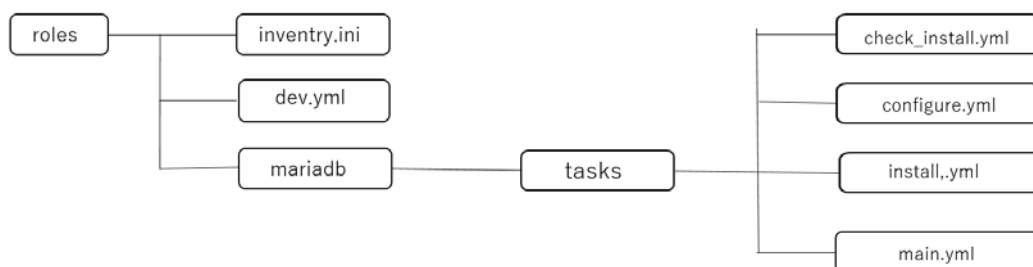


図4 MariDB インストール用ロール構成

付した⁸⁾。Ansible を用いて管理対象サーバ (IP:10.0.2.5) に MariaDB をインストールした。構築環境を図3に示す。管理対象サーバ (IP:10.0.2.5) に MariaDB をインストールするために6個のソースファイルを作成した。これらのロール構成を図4に示す。Ansible は、宛先を記載する inventory ファイル、主な動作を記載する playbook ファイル (図4では dev. yml) を作成し、実行コマンドを実行し、管理対象を管理する。ここでは inventory ファイル、playbook ファイルだけではなく、tasks ディレクトリ以下に4個のファイルにロールを記述し、MariaDB のインストールを行う。ロールとは playbook を分割・構成するコンポーネントであり、再利用性を高めるものである。

(2) 構築結果

コマンド

```
$ dnf module list mariadb
```

により、管理対象サーバにインストールできる MariaDB のバージョンを調べた。2021年4月に構築を開始したので、mariadb10.3をインストールした。本研究では、情報基盤センター演習室のPCのLinux環境を作

め、“mariadb-server”をインストールする。inventory ファイル (図5) には、インストールしたい管理対象サーバのIPアドレスを記述する。本研究では10.0.2.5の仮想PCにインストールする。

```
[dev]
10.0.2.5
```

図5 roles/inventory. ini

playbook ファイル dev. yml (図6) では、roles ディレクティブを使用することで、作成した mariadb ロールを参照している。これで roles/mariadb/tasks に記述したタスクが実行される。また、“become:true”はsudo権限でタスク実行を行う。sudoに必要なパスワードは、PASSWORDと伏せている。ここでは、dev. yml にロールに必要なすべての作業を記述せず、作業を tasks ディレクトリ内に作成したYAMLファイルへ分割して記述することで、分かりやすく整理している。

mariadb ロールのインストール作業のYAMLファイルを図7に示す。install. yml の“Add repository”タス

クでは yum repository モジュールを使用して /etc/yum.repos.d/配下にリポジトリ定義ファイル MariaDB.repo を追加することを示している。実行を終えると、/etc/yum.repos.d/MariaDB.repo が作成される。

```
hosts: dev
vars:
ansible_become_pass : <PASSWORD>
become: true
roles:
- mariadb
```

図 6 MariaDB インストール用 roles/dev.yml

```
name: Add repository
yum_repository:
name: MariaDB
description: MariaDB repo
baseurl: http://yum.mariadb.org/10.3/rhel7-amd64
gpgkey: https://yum.mariadb.org/RPG-GPG-KEY-MariaDB
gpgcheck: true
name: Module install mariadb
yum:
name: mariadb
state: present
- name: Install mariadb server
yum:
name: mariadb-server
state: present
enablerepo: mariadb
-name: Start mariadb service
systemd:
name: mariadb
state: started
enabled: yes
```

図 7 MariaDB インストール用

roles/mariadb/tasks/install.yml

実行コマンドと実行結果を図 8 に示す。“Add repository” タスクに/etc/yum.repos.d/MariaDB.repo が作成されたことが分かる。図 8 より MariaDB がインストールできた。

```
[ansiblecontrol@localhost roles] $ ansible-playbook
-i inventory.ini dev.yml --diff
PLAY [dev] *
TASK [Gathering Facts]
ok: 10.0.2.5
TASK [mariadb : Add repository] ****
- before: etc yum.repos.d/MariaDB.repo +++ after:
/etc/yum.repos.d/MariaDB.repo 2 -0,0 +1,6 @@ +
MariaDB #baseurl = http://yum.
mariadb.org/10.3/rhel7-amd64 #gpgcheck = 1 #gpgkey
= https://yum.mariadb.org/RPG-GPG-KEY-MariaDB
+name = MariaDB repo
changed: [10.0.2.5]
TASK [Module install mariadb] ****
changed: [10.0.2.5]
TASK [mariadb : Install mariadb server] ****
ok: [10.0.2.5]
TASK [mariadb : Start mariadb service]***
changed: [10.0.2.5]
PLAY RECAP***
10.0.2.5:ok changed=3 unreachable=0 gailed=0
skipped=0
rescued=0 ignored=0
```

図 8 MariaDB インストール用 playbook の実行結果

3.2 PHP

(1) 環境構築

Ansible を用いて PHP を管理サーバ(IP:10.0.2.4)と管理対象サーバ(IP:10.0.2.5)にインストールした。図 3 の MariaDB の構成と同様に管理対象サーバ(IP:10.0.2.5)に PHP のインストールを行った。PHP インストール用のロール構成を図 9 に示す。PHP でも MariaDB と同様に、inventory ファイル、playbook ファイルだけではなく、設定ファイル 2 個を記述し、PHP をインストールする。

(2) 構築結果

コマンド

```
$ dnf module list php
```

により、管理対象サーバにインストールできる PHP のバージョンを調べた。2021 年 8 月に構築を開始したので、php7.4 をインストールした。本研究では、情報基盤センター演習室の PC の Linux 環境を作るため、“php-common”、“php-devel”をインストールする。図 10 に playbook ファイル(dev.yml)を示す。Inventory ファイ

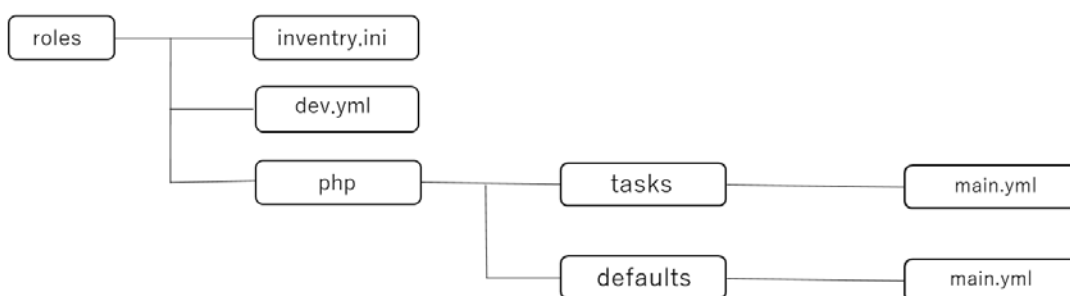


図9 PHP インストール用ロール構成

ルは図5と同じである。

図6では“roles: -mariadb”とすることでmariadbディレクトリ内のタスクを実行した。ここでは，“roles:-php”とすることでPHPディレクトリ内のタスクを実行する。今後，playbookで仮想マシンにgccやbinutils等をインストールしていくので，どのタスクにも対応できるplaybookを作成する。同時に実行したければ“roles:-mariadb -php”のように増やしていけば良い。

```

hosts: dev
vars:
ansible_become_pass : <PASSWORD>
become: true
roles:
- php
    
```

図10 PHP インストール用 roles/dev.yml

PHPのインストール用タスクを記述したファイルを図11に示す。“rpm_key”でRPM(Red Hat Package Manager)パッケージを“state:present”でインストールする。“Add EPEL Repository”でEPELレポジトリをインストール，“Add Remi Repository”でRemiレポジトリのインストールを行う。EPEL(Extra Packages for Enterprise Linux)とは，Red Hat Enterprise LinuxやCentOSで使えるパッケージレポジトリのことである。Remiレポジトリは様々なバージョンのphpのパッケージを配布しているレポジトリである。“Enable PowerTolls”で管理対象側でコマンド“dnf config-manager -set-enabled PowerTolls”を実行して

PowerToolsを有効にする。PowerToolsはもともとデフォルトで無効になっており，EPELパッケージに必要な開発ツールが含まれているためこれを有効にする。“Install php”で管理対象側でコマンド“dnf module install -y php:remi-{{ php_version }}"を実行してPHPをインストールする。

“Install Modules”でインストールするPHPモジュールを指定する。本研究では，情報基盤センター演習室のPCと同じ環境にするために“php-common”と“php-devel”を指定してインストールする。“Configure php”でphpプログラムの全体的な動作環境を設定するファイル(phi.ini)を作成する。“with_items”の中身を以下に説明する。

- “mbstring”：PHPでマルチバイト文字を使用できるようにするPHPの拡張モジュールのことで，mbstringの設定を行うことで，PHP内で日本語が使えるようになる。
- “post_max_size”：POSTデータの許容最大サイズを設定する。
- “upload_max_filesize”：アップロードファイルの許容サイズを設定する。
- “mbstring.language”：デフォルトの言語設定する。
- “mbstring.internal_encoding”：内部文字エンコーディングの変更・確認する。
- “mbstring.encoding_translation”：HTTP入力変換の設定する。
- “mbstring.detect_order”：文字コードの検出順を指定する。

図12には，PHPのデフォルト値を記述した。図13にPHPのインストールのコマンドと実行結果を示す。

```

rpm_key:
state: present
key: https://rpms. remirepo. net/RPM-GPG-KEY-remi2018
rpm key:
state: present
key: https://dl. fedoraproject. org/pub/epel/RPM-GPG-KEY-EPEL-8
name: Add EPEL Repository dnt:
name: "https://dl. fedoraproject. org/pub/epel/epel-release-latest-8. noarch. rpm"state: present
name: Add Remi Repository
dnf:
name: "http://rpms. remirepo. net/enterprise/remi-release-8. rpm
state: present
name: Enable Power Tools
shell:
cmd: dnf config-manager --set-enabled Power Tools
name: Install php
shell:
cmd: dnf module install -y php: remi- {{ php_version }}
-name: Install Modules
dnf:
name: ['php-common', 'php-devel']
state: present
name: Configure php ini_file:
path: /etc/php. ini section: "{{item. section }}" option: "{{ item. option }}" value: "{{item. value }}"
backup: "{{ item. backup }}"
with_items:
- section: php, option: post_max_size, value: "{{ filesize }}" , backup: true } - { section: php, option:
upload_max_filesize, value: "{{ filesize }}" , backup: fals
- { section: Date, option: date. timezone, value: "{{ timezone }}" , backup: false }
- { section: mbstring, option: mbstring. Language, value: " language }}" , backup: false }
- { section: mbstring, option: mbstring internal_encoding, value: "{{ internal_enco ding }}" , backup: false }
- { section: _mbstring, option: mbstring. encoding translation, value: "{{ encoding. t ranslation }}" , backup:
false }
- { section: mbstring, option: mbstring. detect_order, value: "{{ detect_order }}" . backup: false }

```

図 11 roles/php/tasks/main. yml

```

[ansiblecontrol@localhost roles] $ ansible-playbook -i inventory. ini dev. yml --diff
PLAY [dev] *****
(省略)
before: etc/php. ini content) +++ after: /etc/php. ini (content)
mbstring) mbstring. detect_order = auto
changed: (10. 0. 2. 5) => (item={'section': 'mbstring', 'option': 'mbstring. detect_order' 'value': 'auto',
'backup': False})

```

図 13 PHP インストール用 playbook の実行結果

```

—
php_version: 7.4
language: Japanese
timezone: Asia Tokyo
internal_encoding: UTF-8
encoding_translation: off
detect_order: auto
filesize: 128M

```

図 12 roles/php/defaults/main.yml

4. 結論

本研究では、構成管理ツール Ansible を用い、本校の情報基盤センター演習室の PC 管理の省力化を行うことを目的とした。そのために、Ansible を用いて仮想マシンに MariaDB と PHP のインストールを行った。今後は、同様に情報基盤センターの演習室の PC と同じ環境構築を行う。MariaDB と PHP のほかに binutils, open JDK 等のソフトウェアインストールを行う。現在は binutils のインストールに向けて playbook に記述する内容を考慮中である。

参考文献

- 1) 北山 晋吾ほか, Ansible 実践ガイド第3版, インプレス(2019)
- 2) CentOS, Download, <https://www.centos.org/download/> (2021) (閲覧日 2020 年 4 月 10 日)
- 3) Oracle VM VirtualBox, <https://www.virtualbox.org/> (2022) (閲覧日 2022 年 4 月 10 日)
- 4) RedHat, Ansible, <https://www.ansible.com/> (閲覧日 2020 年 4 月 16 日)
- 5) MariaDB Foundation, MariaDB.org, <https://mariadb.org/> (2021) (閲覧日 2021 年 4 月 6 日)
- 6) PHP, <https://windows.php.net/download/> (2022), (閲覧日 2021 年 10 月 18 日)
- 7) Tomcky' s blog, [入門・初心者向け]Ansible で MariaDB(MySQL) をインストールして起動するところまで, <https://tomcky.hatenadiary.jp/entry/20190513/1557707578> (2019) (閲覧日 2021 年 4 月 6 日)
- 8) 小畑 沙生, Ansible を用いた演習室 PC の管理作業の省力化～ファイル配布自動化～, 2020 香川