# Ansible紹介

2016.3.1

Ｒ＆Ｄセンター　OSS戦略企画室
OSS技術第二課
角馬　文彦

※本文中の会社名、商品名は、各社の商標及び登録商標です。

# Ansibleについて

- 概要

いわゆる構成管理ツール。

リモートホストに対して特定の言語で指定された

タスクを実行する。

同様のツールとしては chef, puppetなどが有名。

- ➢ システムの構成管理
- ➢ アプリケーションの展開、実行
- ➢ マルチノードオーケストレーション

- 特徴
  - Python ベース
  - エージェントレス
  - リモートホストには標準ではpython以外の環境は必要ない
  - モジュールの作成はPython以外でも可能
    (特定の条件を満たす言語)
  - 作業はシーケンシャルに実行
- ライセンス

  GPL version 3

## ●コード

- ➢ 最新バージョン2.0(Over the Hills and Far Away)
  現在も開発継続中
  ひとつ前1.9.4(Dancing In the Streets)
- ➢ https://pypi.python.org/pypi/ansible
  2.0のものはpypiから取得できる。
- ➢ https://github.com/ansible/ansible
  開発リポジトリ
- ➢ http://releases.ansible.com/ansible/
  リリースターボール

# 動作確認環境

- Ubuntu 14.04
  サーバ、クライアントホスト
- Python 2.7
- ubuntu ansible パッケージ使用
- ansible パッケージ詳細
  次ページ

```
$ apt show ansible
Package: ansible
Priority: optional
Section: universe/admin
Installed-Size: 2,758 kB
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Janos Guljas <janos@debian.org>
Version: 1.5.4+dfsg-1
Depends: python (>= 2.7), python (<< 2.8), python:any (>= 2.7.1-0ubuntu2), python-crypto,
python-yaml, openssh-client | python-paramiko, python-jinja2, python-httplib2
Suggests: ansible-doc, sshpass
Download-Size: 418 kB
Homepage: http://ansible.com
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Origin: Ubuntu
APT-Manual-Installed: yes
APT-Sources: http://us.archive.ubuntu.com/ubuntu/ trusty/universe amd64 Packages
Description: Configuration management, deployment, and task execution system
 Ansible is a radically simple model-driven configuration management,
 multi-node deployment, and remote task execution system. Ansible works
 over SSH and does not require any software or daemons to be installed
 on remote nodes. Extension modules can be written in any language and
 are transferred to managed machines automatically.

N: There is 1 additional record. Please use the '-a' switch to see it
$
```

## ➢ssh 環境は必須

処理の実行には ssh を利用。これがansibleがエージェントレスである理由。その為ssh 環境は必須。

- ✓ リモートホストには ssh でログインできるように設定
  予めパブリックキーを置いておく。
- ✓ パスフレーズの入力待ちが起こらないように設定
  ssh-agent などで回避。

# inventoryファイル

リモートホストはinventoryファイルで指定する。
書式は下記。

```
host-name1
host-ip
[host-group1]
host-name2

[host-group2]
host-name3
```

詳細は下記参照。
http://docs.ansible.com/ansible/intro_inventory.html

# ホストの指定

リモートホストは下記の方法で指定可能。

- /etc/ansible/hosts
- 環境変数

  ANSIBLE_HOSTS=file-path
- オプション

  コマンドのオプションで指定

  ```
  ansible -i file-path ～
  ansible-playbook -i file-path ～
  ```

# ホストの指定例

```
$ cat ./hosts
test-misc1
[testserver1]
tnet-misc2
tnet-misc3
[testserver2]
tnet-misc2
172.17.61.11
$ ansible -i hosts test-misc1 --list-hosts
    test-misc1
$ ansible -i hosts testserver1 --list-hosts
    tnet-misc2
    tnet-misc3
$ ansible -i hosts testserver2 --list-hosts
    tnet-misc2
    172.17.61.11
$
```

# 動作確認用のホスト構成

以降の確認環境では下記のinventoryファイルを利用するものとする。

```
$ cat /etc/ansible/hosts
tnet-misc2
[testserver1]
tnet-misc3
[testserver2]
tnet-misc3
tnet-misc4
$
```

# Ad hoc な実行

下記の書式で直接コマンドラインで実行。

ansible host-name –m module-name [-a options]

ping モジュールを使った例を示す。

```
$ ansible testserver2 -m ping
tnet-misc4 | success >> {
    "changed": false,
    "ping": "pong"
}

tnet-misc3 | success >> {
    "changed": false,
    "ping": "pong"
}

$
```

```
$ ansible all -m ping
tnet-misc2 | success >> {
    "changed": false,
    "ping": "pong"
}

tnet-misc3 | success >> {
    "changed": false,
    "ping": "pong"
}

tnet-misc4 | success >> {
    "changed": false,
    "ping": "pong"
}

$
```

ping モジュールは pingコマンドでは無い。

(ICMP を送信したりはしない。)

# 任意のコマンドの実行

下記の書式で任意のコマンドを実行可能。

```
ansible host-name -a command-line
```

uname コマンドを使った例を示す。

```
$ ansible testserver2 -a 'uname -a'
tnet-misc3 | success | rc=0 >>
Linux tnet-misc3 3.16.0-30-generic #40~14.04.1-Ubuntu SMP Thu Jan 15 17:43:14
UTC 2015 x86_64 x86_64 x86_64 GNU/Linux

tnet-misc4 | success | rc=0 >>
Linux tnet-misc4 3.16.0-30-generic #40~14.04.1-Ubuntu SMP Thu Jan 15 17:43:14
UTC 2015 x86_64 x86_64 x86_64 GNU/Linux

$
```

# PlayBooks

- PlayBooks

  処理をansibleに指示する為の言語

- PlayBook

  PlayBooks 言語で書かれたファイル

- 形式

  PlayBooks はYAML フォーマットで表現。

- PlayBookの実行

  ansible-playbook playbook-file-name [options]

# PlayBookの書式

playbook の基本フォーマットを示す。

```
- hosts: host-name
  remote_user: user-name
  tasks:
    - name: task-name
      module-name: module-options
```

以降では playbook の良く使いそうな機能の例を示す。

# ➢ モジュールの実行

```
$ cat test-ping.yml
- hosts: tnet-misc2
  user: ubuntu
  tasks:
    - name: check host
      ping:
$ ansible-playbook test-ping.yml

PLAY [tnet-misc2] **************************************************************

GATHERING FACTS ****************************************************************
ok: [tnet-misc2]

TASK: [check host] *************************************************************
ok: [tnet-misc2]

PLAY RECAP *********************************************************************
tnet-misc2                 : ok=2    changed=0    unreachable=0    failed=0

$
```

# ➢ 任意のコマンドの実行

```
$ cat test-uname-cmd.yml
- hosts: tnet-misc2
  user: ubuntu
  tasks:
    - name: display results of uname
      command: uname -a
$ ansible-playbook test-uname-cmd.yml

PLAY [tnet-misc2] *************************************************************

GATHERING FACTS ***************************************************************
ok: [tnet-misc2]

TASK: [display results of uname] *************************************************
changed: [tnet-misc2]

PLAY RECAP ************************************************************************
tnet-misc2                 : ok=2    changed=1    unreachable=0    failed=0

$
```

# ➢ 実行結果の表示１(--verbose オプション)

```
$ ansible-playbook test-uname-cmd.yml --verbose

PLAY [tnet-misc2] **************************************************************

GATHERING FACTS ***************************************************************
ok: [tnet-misc2]

TASK: [display results of uname] ***********************************************
changed: [tnet-misc2] => {"changed": true, "cmd": ["uname", "-a"], "delta": "0:00:00.001116", "end": "2016-02-12 04:53:42.664070", "rc": 0, "start": "2016-02-12 04:53:42.662954", "stderr": "", "stdout": "Linux tnet-misc2 3.16.0-30-generic #40~14.04.1-Ubuntu SMP Thu Jan 15 17:43:14 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux"}

PLAY RECAP ********************************************************************
tnet-misc2                 : ok=2    changed=1    unreachable=0    failed=0

$
```

## ➢ 実行結果の表示2(debug モジュール)

```
$ cat test-uname-dbg.yml
- hosts: tnet-misc2
  user: ubuntu
  tasks:
    - name: display results of uname
      command: uname -a
      register: uname_res

    - debug: var=uname_res.stdout_lines
$
```

```
$ ansible-playbook test-uname-dbg.yml

PLAY [tnet-misc2] *************************************************************

GATHERING FACTS ***************************************************************
ok: [tnet-misc2]

TASK: [display results of uname] **********************************************
changed: [tnet-misc2]

TASK: [debug var=uname_res.stdout_lines] **************************************
ok: [tnet-misc2] => {
    "uname_res.stdout_lines": [
        "Linux tnet-misc2 3.16.0-30-generic #40~14.04.1-Ubuntu SMP Thu Jan 15 17:43:14 UTC 2015
x86_64 x86_64 x86_64 GNU/Linux"
    ]
}

PLAY RECAP ********************************************************************
tnet-misc2                 : ok=3    changed=1    unreachable=0    failed=0

$
```

## ➢ プロンプト

```
$ cat test-prompt.yaml
- user: ubuntu
  hosts: tnet-misc2
  vars:
    file_path: /home/ubuntu/config
  vars_prompt:
    - name: "inputdata"
      prompt: "Please enter data"
      private: no                                                          1)
      default: "test data"
  tasks:
    - lineinfile: dest={{ file_path }} create=yes line="{{ inputdata }}"
    - command: cat {{ file_path }}
      register: cat_results
    - debug: var=cat_results.stdout_lines
$
```

1) yes の場合、入力値のエコーバックを行わない。

```
$ ansible-playbook test-prompt.yaml
Please enter data [test data]: 192.168.1.1 host

PLAY [tnet-misc2] ***************************************************************

GATHERING FACTS *****************************************************************
ok: [tnet-misc2]

TASK: [lineinfile dest=/home/ubuntu/config create=yes line="192.168.1.1 host"] ***
changed: [tnet-misc2]

TASK: [command cat /home/ubuntu/config] ***************************************
changed: [tnet-misc2]

TASK: [debug var=cat_results.stdout_lines] ************************************
ok: [tnet-misc2] => {
    "cat_results.stdout_lines": [
        "192.168.1.1 host"
    ]
}

PLAY RECAP **********************************************************************
tnet-misc2                 : ok=4    changed=2    unreachable=0    failed=0

$
```

## ➢ タスクの実行エラー

下記の playbook を使用した時のエラー例を示す。

```
$ cat test-lynx-list.yml
- hosts: testserver2
  user: ubuntu
  sudo: yes
  tasks:
    - name: list lynx
      command: dpkg -l lynx
$
```

# ホスト shutdown 中

```
$ ansible-playbook test-lynx-list.yml

PLAY [testserver2] ********************************************************************

GATHERING FACTS ***********************************************************************
fatal: [tnet-misc4] => SSH encountered an unknown error during the connection. We recommend you
re-run the command using -vvvv, which will enable SSH debugging output to help diagnose the issue
fatal: [tnet-misc3] => SSH encountered an unknown error during the connection. We recommend you
re-run the command using -vvvv, which will enable SSH debugging output to help diagnose the issue

TASK: [list lynx] *********************************************************************
FATAL: no hosts matched or all hosts have already failed -- aborting


PLAY RECAP *****************************************************************************
           to retry, use: --limit @/home/ubuntu/test-lynx-list.retry

tnet-misc3                 : ok=0    changed=0    unreachable=1    failed=0
tnet-misc4                 : ok=0    changed=0    unreachable=1    failed=0

$ cat /home/ubuntu/test-lynx-list.retry
tnet-misc3
tnet-misc4
$
```

# コマンド実行エラー

```
$ ansible-playbook test-lynx-list.yml

PLAY [testserver2] ***************************************************************

GATHERING FACTS ******************************************************************
ok: [tnet-misc4]
ok: [tnet-misc3]

TASK: [list lynx] ***************************************************************
changed: [tnet-misc3]
failed: [tnet-misc4] => {"changed": true, "cmd": ["dpkg", "-l", "lynx"], "delta": "0:00:00.008187", "end":
"2016-02-17 02:54:18.511604", "rc": 1, "start": "2016-02-17 02:54:18.503417"}
stderr: dpkg-query: no packages found matching lynx

PLAY RECAP ***********************************************************************
           to retry, use: --limit @/home/ubuntu/test-lynx-list.retry

tnet-misc3                 : ok=2    changed=1    unreachable=0    failed=0
tnet-misc4                 : ok=1    changed=0    unreachable=0    failed=1

$ cat /home/ubuntu/test-lynx-list.retry
tnet-misc4
$
```

# ➢ 複数タスクの処理

```
$ cat test-ovs-install.yml
- hosts: tnet-misc2
  user: ubuntu
  sudo: yes
  tasks:
    - name: install ovs
      apt: name=openvswitch-switch state=latest
    - name: create bridge
      command: ovs-vsctl --may-exist add-br br-test
    - name: check bridge
      command: ovs-vsctl show
      register: ovs_results
    - debug: var=ovs_results.stdout_lines # output result of ovs-vsctl
$
```

```
$ ansible-playbook test-ovs-install.yml

PLAY [tnet-misc2]
***************************************************************


GATHERING FACTS
*****************************************************************
ok: [tnet-misc2]

TASK: [install ovs]
*************************************************************
changed: [tnet-misc2]

TASK: [create bridge]
***********************************************************
changed: [tnet-misc2]

TASK: [check bridge]
************************************************************
changed: [tnet-misc2]
```

```
TASK: [debug var=ovs_results.stdout_lines] ************************************
ok: [tnet-misc2] => {
    "ovs_results.stdout_lines": [
        "132fbd78-de70-422e-93a4-c3d2b7ffd6cf",
        "    Bridge br-test",
        "        Port br-test",
        "            Interface br-test",
        "                type: internal",
        "    ovs_version: ¥"2.0.2¥""
    ]
}

PLAY RECAP
***************************************************************************
tnet-misc2                 : ok=5    changed=3    unreachable=0    failed=0

$
```

- playbook の流用

  include 文を使い、ある playbook に他のplaybookをインポートする事ができる。

  前述のplaybookをインポートした例を示す。

```
$ cat test-main.yml
- include: test-ping.yml
- include: test-uname-cmd.yml
$
```

include は playbook のインポートだけでなく tasks 文で指定して task を取り込むこともできる。

```
$ ansible-playbook test-main.yml

PLAY [testserver2] ***************************************************************

GATHERING FACTS ******************************************************************
ok: [tnet-misc3]
ok: [tnet-misc4]

TASK: [check host] ***************************************************************
ok: [tnet-misc3]
ok: [tnet-misc4]

PLAY [tnet-misc2] ****************************************************************

GATHERING FACTS ******************************************************************
ok: [tnet-misc2]

TASK: [display results of uname] ************************************************
changed: [tnet-misc2]

PLAY RECAP ************************************************************************
tnet-misc2           : ok=2    changed=1    unreachable=0    failed=0
tnet-misc3           : ok=2    changed=0    unreachable=0    failed=0
tnet-misc4           : ok=2    changed=0    unreachable=0    failed=0

$
```

## ➢ もう少し複雑な例

下記の処理を実行

✓ apache, lynx がインストールされていなければインストール

✓ lynx で apache のデフォルトページを表示

(バリエーションの為tnet-misc2 には予め lynx をインストールする。)

```
$ cat test-apache-install.yml
- hosts: all
  user: ubuntu
  sudo: yes
  tasks:
    - name: update package index
      apt: update_cache=yes

- hosts: tnet-misc2
  user: ubuntu
  sudo: yes
  tasks:
    - name: check if apache2 exists
      command: dpkg -l apache2
      ignore_errors: True                                  1)
      register: dpkg_results                               2)
    - name: install apache
      when: dpkg_results|failed                            3)
      apt: name=apache2 state=latest
      notify:                                              4)
        - take interval
  handlers:                                                5)
    - name: take interval
      pause: seconds=3
- hosts: all
  user: ubuntu
  sudo: yes
  tasks:
    - name: check if lynx exists
      command: dpkg -l lynx
      ignore_errors: True
      register: dpkg_results
    - name: install lynx
      when: dpkg_results|failed
      apt: name=lynx state=latest
```

```
- hosts: testserver2
  user: ubuntu
  sudo: yes
  tasks:
    - name: diaplay apache init page
      command: lynx -dump http://tnet-misc2
      ignore_errors: True
      register: lynx_results
    - debug: var=lynx_results.stdout_lines # output dump data of lynx

- hosts: all
  user: ubuntu
  sudo: yes
  tasks:
    - name: verify packages after install
      command: dpkg -l apache2 lynx
      ignore_errors: True
      register: dpkg_results
    - debug: var=dpkg_results.stdout_lines # check if apache2 and lynx are installed
$
```

1) 実行結果を無視
2) 実行結果を指定の変数に格納
3) 条件判定
4) ハンドラ定義
5) ハンドラの指定

```
$ ansible tnet-misc2 --sudo -m apt -a 'name=lynx state=latest'
tnet-misc2 | success >> {
    "changed": true,
    "stderr": "",
    "stdout": "Reading package lists...¥nBuilding dependency tree...¥nReading state
information...¥nThe following extra packages will be installed:¥n  lynx-cur¥nThe
following NEW packages will be installed:¥n  lynx lynx-cur¥n0 upgraded, 2 newly installed,
0 to remove and 151 not upgraded.¥nNeed to get 960 kB of archives.¥nAfter this
operation, 2570 kB of additional disk space will be used.¥nGet:1
http://us.archive.ubuntu.com/ubuntu/ trusty/main lynx-cur amd64 2.8.8pre4-1 [956
kB]¥nGet:2 http://us.archive.ubuntu.com/ubuntu/ trusty/main lynx all 2.8.8pre4-1 [4184
B]¥nFetched 960 kB in 1s (689 kB/s)¥nSelecting previously unselected package lynx-
cur.¥n(Reading database ... 61559 files and directories currently installed.)¥nPreparing to
unpack .../lynx-cur_2.8.8pre4-1_amd64.deb ...¥nUnpacking lynx-cur (2.8.8pre4-
1) ...¥nSelecting previously unselected package lynx.¥nPreparing to
unpack .../lynx_2.8.8pre4-1_all.deb ...¥nUnpacking lynx (2.8.8pre4-1) ...¥nProcessing
triggers for mime-support (3.54ubuntu1.1) ...¥nProcessing triggers for man-db (2.6.7.1-
1ubuntu1) ...¥nSetting up lynx-cur (2.8.8pre4-1) ...¥nupdate-alternatives: using
/usr/bin/lynx to provide /usr/bin/www-browser (www-browser) in auto mode¥nSetting
up lynx (2.8.8pre4-1) ...¥n"
}

$
```

```
$ ansible-playbook test-apache-install.yml

PLAY [all] ***************************************************************************************

GATHERING FACTS
*********************************************************************************
ok: [tnet-misc2]
ok: [tnet-misc3]
ok: [tnet-misc4]

TASK: [update package index] ****************************************************************
ok: [tnet-misc3]
ok: [tnet-misc2]
ok: [tnet-misc4]

PLAY [tnet-misc2] ******************************************************************************

GATHERING FACTS
*********************************************************************************
ok: [tnet-misc2]
```

```
TASK: [check if apache2 exists] ***********************************************
failed: [tnet-misc2] => {"changed": true, "cmd": ["dpkg", "-l", "apache2"], "delta":
"0:00:00.007453", "end": "2016-02-14 03:28:59.179386", "rc": 1, "start": "2016-02-14
03:28:59.171933"}
stderr: dpkg-query: no packages found matching apache2
...ignoring

TASK: [install apache] ************************************************************
changed: [tnet-misc2]

NOTIFIED: [take interval] *********************************************************
(^C-c = continue early, ^C-a = abort)
[tnet-misc2]
Pausing for 3 seconds
ok: [tnet-misc2]

PLAY [all] *************************************************************************

GATHERING FACTS
*******************************************************************
ok: [tnet-misc4]
ok: [tnet-misc3]
ok: [tnet-misc2]
```

```
TASK: [check if lynx exists] ****************************************************
failed: [tnet-misc4] => {"changed": true, "cmd": ["dpkg", "-l", "lynx"], "delta":
"0:00:00.008070", "end": "2016-02-14 03:29:27.453067", "rc": 1, "start": "2016-02-14
03:29:27.444997"}
stderr: dpkg-query: no packages found matching lynx
...ignoring
failed: [tnet-misc3] => {"changed": true, "cmd": ["dpkg", "-l", "lynx"], "delta":
"0:00:00.008109", "end": "2016-02-14 03:29:27.450850", "rc": 1, "start": "2016-02-14
03:29:27.442741"}
stderr: dpkg-query: no packages found matching lynx
...ignoring
changed: [tnet-misc2]

TASK: [install lynx] *************************************************************
skipping: [tnet-misc2]                    --> 処理のスキップ
changed: [tnet-misc4]
changed: [tnet-misc3]

PLAY [testserver2] ***************************************************************

GATHERING FACTS
******************************************************************
ok: [tnet-misc4]
ok: [tnet-misc3]
```

```
TASK: [diaplay apache init page] **********************************************
changed: [tnet-misc4]
changed: [tnet-misc3]

TASK: [debug var=lynx_results.stdout_lines] ***********************************
ok: [tnet-misc3] => {
    "lynx_results.stdout_lines": [
        "   Ubuntu Logo Apache2 Ubuntu Default Page",
        "   It works!",
        "",
        "   This is the default welcome page used to test the correct operation of",
        "   the Apache2 server after installation on Ubuntu systems. It is based on",
        "   the equivalent page on Debian, from which the Ubuntu Apache packaging",
途中省略
    ]
}
ok: [tnet-misc4] => {
    "lynx_results.stdout_lines": [
        "   Ubuntu Logo Apache2 Ubuntu Default Page",
途中省略
        "  10. http://validator.w3.org/check?uri=referer"
    ]
}
```

```
PLAY [all] *******************************************************************************

GATHERING FACTS *************************************************************************
ok: [tnet-misc2]
ok: [tnet-misc4]
ok: [tnet-misc3]

TASK: [verify packages after install] ***********************************************
failed: [tnet-misc3] => {"changed": true, . . .
途中省略
failed: [tnet-misc4] => {"changed": true, . . .
途中省略
changed: [tnet-misc2]

TASK: [debug var=dpkg_results.stdout_lines] ****************************************
ok: [tnet-misc2] => {
    "dpkg_results.stdout_lines": [
        "Desired=Unknown/Install/Remove/Purge/Hold",
        "| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend",
        "|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)",
        "||/ Name                                Version                             Architecture Description",
        “+++-======================================-.  .  . ==",
        "ii  apache2                             2.4.7-1ubuntu4.9              amd64       Apache HTTP Server",
        "ii  lynx                                2.8.8pre4-1                   all         Text-mode WWW Browser (transitional
package)"
```

```
   ]
}
ok: [tnet-misc3] => {
   "dpkg_results.stdout_lines": [
      "Desired=Unknown/Install/Remove/Purge/Hold",
      "| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend",
      "|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)",
      "||/ Name                        Version                   Architecture Description",
      “+++-===================================-.  .  .  ==",
      "ii  lynx                     2.8.8pre4-1               all        Text-mode WWW Browser (transitional
package)"
   ]
}
ok: [tnet-misc4] => {
途中省略
}

PLAY RECAP ************************************************************************
tnet-misc2              : ok=11   changed=4    unreachable=0    failed=0
tnet-misc3              : ok=11   changed=4    unreachable=0    failed=0
tnet-misc4              : ok=11   changed=4    unreachable=0    failed=0

$
```

# 参考資料

http://docs.ansible.com/

http://docs.ansible.com/intro_getting_started.html

http://docs.ansible.com/ansible/modules.html

http://docs.ansible.com/ansible/modules_by_category.html

https://github.com/ansible/ansible-examples