

# Serverspec ハンズオンテキスト

## **Lesson 1**

## 目次

1	前提条件	4
2	当日の準備	4
3	ハンズオン時の注意	4
4	ハンズオンの概要とゴール	4
5	ハンズオン実習	5
5.1	ハンズオンマシンにログイン	5
5.2	Serverspec Target にログイン	6
5.3	Serverspec Host にログイン	7
5.4	Serverspec の初期化	8
5.5	Serverspec Spec ファイルの編集	9
5.6	Serverspec テストの実行（コンフィグ前）	10
5.7	Ansible による Wordpress 環境自動構築	11
5.8	Serverspec テストの実行（コンフィグ後）	12
5.9	WordPress サイトにアクセス	12
6	おまけ	13
6.1	やり直したい時	13
7	参考	13

## 改訂履歴

版数	年月日	概要	作成者
1.0	2016 年 7 月 3 日	初版	倉持 健史
1.1	2016 年 7 月 4 日	BugFix	倉持 健史

## 1 前提条件

本ハンズオンには、まず **Ansible の理解が必要**です。Ansible ハンズオンの Lesson1 か Lesson2 を受講していない方はできるだけ先にそちらを受講ねがいます。既に同程度の知識を有するのであればその限りではありません。

## 2 当日の準備

ハンズオンを行うために必要な環境は以下です。

- TETRA 環境につながるネットワーク環境
- リモートログインするための端末（パソコンなど）
- ssh でログインして行うため、SSH クライアント（例：Teraterm、openssh-clients など）

## 3 ハンズオン時の注意

本環境はコンテナ環境で作成されていますが、受講者はコンテナ環境であることは意識せずに Serverspec の本質を的確にハンズオンすることができます。なお、手動でコンテナを操作することを簡易的に抑止していますが、直接操作は行わないようにしてください。環境の再構築などは、本書の「6 おまけ」で別途用意しているツールを利用するようにしてください。

## 4 ハンズオンの概要とゴール

本ハンズオンを通じて Serverspec の簡単な利用方法を体験することができます。ハンズオンでは、2つのマシン Serverspec Host、Serverspec Target を利用します。まず、Target マシンを何も無い状態から、Ansible により Wordpress の初期セットアップが完了した状態にします。その前後で Serverspec を使ってマシンの状態をテストしその結果を比較を通じて利用方法の理解いただく事をゴールとします。

## 5 ハンズオン実習

### 5.1 ハンズオンマシンにログイン

自身のアカウントでハンズオン環境が用意されているマシンに SSH クライアントを使ってログインします。ログイン後、下記のようなディレクトリが保存されていることを確認します。

Notes

アカウント/パスワード、ハンズオンマシンの **IP アドレス** は実習時にお知らせします。

```
# ssh takeshik@192.168.0.123
takeshik@192.168.0.123's password:
Last login: Sat Apr 23 18:34:53 2016
[takeshik@handson ~]$ ls -F
lesson1_files/
```

Notes

[~/lesson1\\_files/tools](#) 以下はハンズオンそのもには必要ありません。用途は本書「6 おまけ」に記載しています。

## 5.2 Serverspec Target にログイン

Serverspec Target マシンにログインします。ログインするためのシェルスクリプトが用意されているので以下のように引数を指定してログインします。

### Serverspec Target へログインする場合

```
[takeshik@handson ~]$ ./lesson1_files/login.sh target  
  
Login Serverspec Target...  
  
[root@target ~] $
```

Target の IP アドレスを確認しておきます。後で使うため、記憶しておくかメモしておいてください。

Notes : IP アドレスの確認

```
$ ip addr show eth0  
104: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default  
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff  
    inet 172.17.0.3/16 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::42:acff:fe11:2/64 scope link  
        valid_lft forever preferred_lft forever
```

上記の場合は、172.17.0.3 です。

Notes

ログイン後は、スーパーユーザー (root) になっています。今回の環境のオペレーションはスーパーユーザーで行います。パスワードは「root」です。

### 5.3 Serverspec Host にログイン

Serverspec Host マシンにログインします。ログインするためのシェルスクリプトが用意されているので以下のよう  
に引数を指定してログインします。

#### Serverspec Host へログインする場合

```
[takeshik@handson ~]$ ./lesson1_files/login.sh host  
  
Login Serverspec Host...  
  
[root@host ~] $
```

ログイン後に、Target に Host の SSH 公開鍵を渡しておきます。ここで先ほどの Target の IP アドレスを使いま  
す。

```
[root@host ~] $ ssh-copy-id 172.17.0.3
```

#### Notes

スーパーユーザー (root) のパスワードは「root」です。

きちんと渡せたか、以下のコマンドで確認します。

```
[root@host ~] $ ssh root@172.17.0.3 ls /root/  
anaconda-ks.cfg  
install.log  
install.log.syslog
```

このように Target 側の root ディレクトリ内がリストされれば OK です。

#### Notes

ログイン後は、スーパーユーザー (root) になっています。今回の環境のオペレーションはスーパーユーザーで  
行います。パスワードは「root」です。

## 5.4 Serverspec の初期化

Serverspec Host にログインした後に以下のようにして Serverspec の初期化を行います。途中「Input target host name :」では先ほど確認した Target の IP アドレスを入力します。

```
[takeshik@handson ~]$ ./lesson1_files/login.sh host

Login Serverspec Host...

[root@host ~] $ cd Serverspec

[root@host Serverspec] $ serverspec-init
Select OS type:

    1) UN*X
    2) Windows

Select number: 1

Select a backend type:

    1) SSH
    2) Exec (local)

Select number: 1

Vagrant instance y/n: n
Input target host name: 172.17.0.3
+ spec/
+ spec/172.17.0.3/
+ spec/172.17.0.3/sample_spec.rb
+ spec/spec_helper.rb
+ Rakefile
+ .rspec

[root@host Serverspec] $
```



## 5.5 Serverspec Spec ファイルの編集

初期化の後は sample のスペックファイル (./spec/< Target の IP アドレス>/sample\_spec.rb) ができているのでここではそちらを編集します。中身は以下のようになります。

```
[root@host Serverspec] $ cat spec/172.17.0.3/sample_spec.rb
require 'spec_helper'

describe package('php'), :if => os[:family] == 'redhat' do
  it { should be_installed }
end

describe package('httpd'), :if => os[:family] == 'redhat' do
  it { should be_installed }
end

describe package('mysql-server'), :if => os[:family] == 'redhat' do
  it { should be_installed }
end

describe service('httpd'), :if => os[:family] == 'redhat' do
  it { should be_enabled }
  it { should be_running }
end

describe service('mysqld'), :if => os[:family] == 'redhat' do
  it { should be_enabled }
  it { should be_running }
end

describe file('/var/www/html/wordpress/wp-includes/class-http.php') do
  its(:content) { should match /'http_request_timeout', 20/ }
end

describe port(80) do
  it { should be_listening }
end
```

以下が今回のテストの要約です。

- php , httpd , mysql-server がインストールされているか
- httpd が有効で、起動しているか
- mysqld が有効で、起動しているか
- /var/www/html/wordpress/wp-includes/class-http.php に「'http\_request\_timeout', 20」が含まれてるか
- 80 ポートが LISTEN されているか

## 5.6 Serverspec テストの実行（コンフィグ前）

Spec ファイルを作成したらテストを実行してみます。以下のコマンドで実行します。実行するディレクトリは初期化した /root/Serverspec ディレクトリです。以下出力の抜粋です。まだ、Target の方には何も行っていないのでもっと多くの **テスト失敗** の出力が行われるはずです。

```
[root@host Serverspec] $ rake spec

Package "php"
  should be installed (FAILED - 1)

Package "httpd"
  should be installed (FAILED - 2)

Package "mysql-server"
  should be installed (FAILED - 3)

Service "httpd"
  should be enabled (FAILED - 4)
  should be running (FAILED - 5)

Service "mysqld"
  should be enabled (FAILED - 6)
  should be running (FAILED - 7)

File "/var/www/html/wordpress/wp-includes/class-http.php"
  content
    should match /'http_request_timeout', 20/ (FAILED - 8)

Port "80"
  should be listening (FAILED - 9)

Finished in 1.03 seconds (files took 0.79776 seconds to load)
9 examples, 9 failures
```

## 5.7 Ansible による Wordpress 環境自動構築

Ansible のプレイブックファイルは用意されています。インベントリファイルに Target の IP アドレスをセットして実行します。

```
[root@host Serverspec] $ rake spec
[root@host ~] $ cd Ansible/
[root@host Ansible] $ ls
hosts  roles  site.yml
[root@host Ansible] $ more hosts
[target]
172.17.0.3
[root@host Ansible] $ ansible -i hosts target -m ping
172.17.0.3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[root@host Ansible] $ ansible-playbook -i hosts site.yml

PLAY *****

TASK [wordpress : Install Packages] *****
:
:
:

PLAY RECAP *****
172.17.0.3          : ok=10    changed=9    unreachable=0    failed=0

[root@host Ansible] $
```

## 5.8 Serverspec テストの実行（コンフィグ後）

再度 Serverspec によるテストを行います。

```
[root@host Serverspec] $ rake spec
/usr/bin/ruby -I/usr/local/share/gems/gems/rspec-core-3.5.0/lib:
/usr/local/share/gems/gems/rspec-support-3.5.0/lib
/usr/local/share/gems/gems/rspec-core-3.5.0/exe/ rspec--pattern spec/172.17.0.3/\*_spec.rb

Package "php"
  should be installed

Package "httpd"
  should be installed

Package "mysql-server"
  should be installed

Service "httpd"
  should be enabled
  should be running

Service "mysqld"
  should be enabled
  should be running

File "/var/www/html/wordpress/wp-includes/class-http.php"
  content
    should match /'http_request_timeout', 20/

Port "80"
  should be listening

Finished in 0.44387 seconds (files took 0.71277 seconds to load)
9 examples, 0 failures

[root@host Serverspec] $
```

今回はエラーなく、すべて正常（All Green）となっています。

## 5.9 WordPress サイトにアクセス

ブラウザを通じて WordPress サイトにアクセスしてみましょう。以下の URL を入力して確認してください。

[http://<ハンズオンマシンの IP アドレス>/<自身の ID >\\_wordpress/](http://<ハンズオンマシンの IP アドレス>/<自身の ID >_wordpress/)

正しくアクセスできていれば。完了です。

## 6 おまけ

### 6.1 やり直したい時

今回の環境を再度試したい場合のツールを用意しています。既存の Ansible Host と Target をリセットしたい場合は以下のようにツールを実行します。

```
[takeshik@handson ~]$ ./lesson1_files/tools/setup.sh

=== USAGE ===

ex)
    # ./setup.sh [up|remove|restart] [-v]

Description
    up       : Bootup containers
    stop     : Stop containers
    delete  : Delete containers
    restart  : Delete and Bootup containers

[takeshik@handson ~]$ ./lesson2_files/tools/setup.sh restart

Restarting ...

PLAY *****

TASK [remove ansible target container] *****
ok: [127.0.0.1]

TASK [remove ansible host container] *****
ok: [127.0.0.1]

PLAY RECAP *****
127.0.0.1          : ok=2    changed=0    unreachable=0    failed=0

PLAY *****

TASK [boot ansible target container] *****
changed: [127.0.0.1]

TASK [boot ansible host container] *****
changed: [127.0.0.1]

PLAY RECAP *****
127.0.0.1          : ok=2    changed=2    unreachable=0    failed=0

=====

Try the link below after executing the ansible playbook.

http://192.168.0.123/takeshik_wordpress/ .

=====
```

実習時の Wordpress 自動構築後は、最後に出力されているアドレスを使ってブラウザで確認できます。

## 7 参考

- <http://serverspec.org/>
- <http://docs.ansible.com/>
- <https://wordpress.org/>