

Serverspec ハンズオンテキスト (Web Console 版)

Lesson 1

目次

1	環境メモ	3
2	ハンズオンの概要とゴール	4
3	ハンズオン実施	5
3.1	Serverspec Host と Serverspec Target にログイン	5
3.2	Serverspec Target にログイン	6
3.3	Serverspec Host にログイン	7
3.4	Serverspec の初期化	8
3.5	Serverspec Spec ファイルの編集	9
3.6	Serverspec テストの実行（Ansible による自動設定前）	10
3.7	Ansible による Wordpress 環境自動構築	11
3.8	Serverspec テストの実行（コンフィグ後）	12
3.9	WordPress サイトにアクセス	12
4	参考	13

1 環境メモ

以下はハンズオン時に講師から案内された内容を記入しておくための欄です。

ハンズオンマシン IP アドレス	(当日、講師からの案内があります)
Serverspec Host へのアクセス	http://<ハンズオンマシン IP アドレス>:<ポート番号>/wetty/ssh/root/
Serverspec Target へのアクセス	http://<ハンズオンマシン IP アドレス>:<ポート番号>/wetty/ssh/root/
WordPress へのアクセス	http://<ハンズオンマシン IP アドレス>:<ポート番号>/wordpress

2 ハンズオンの概要とゴール

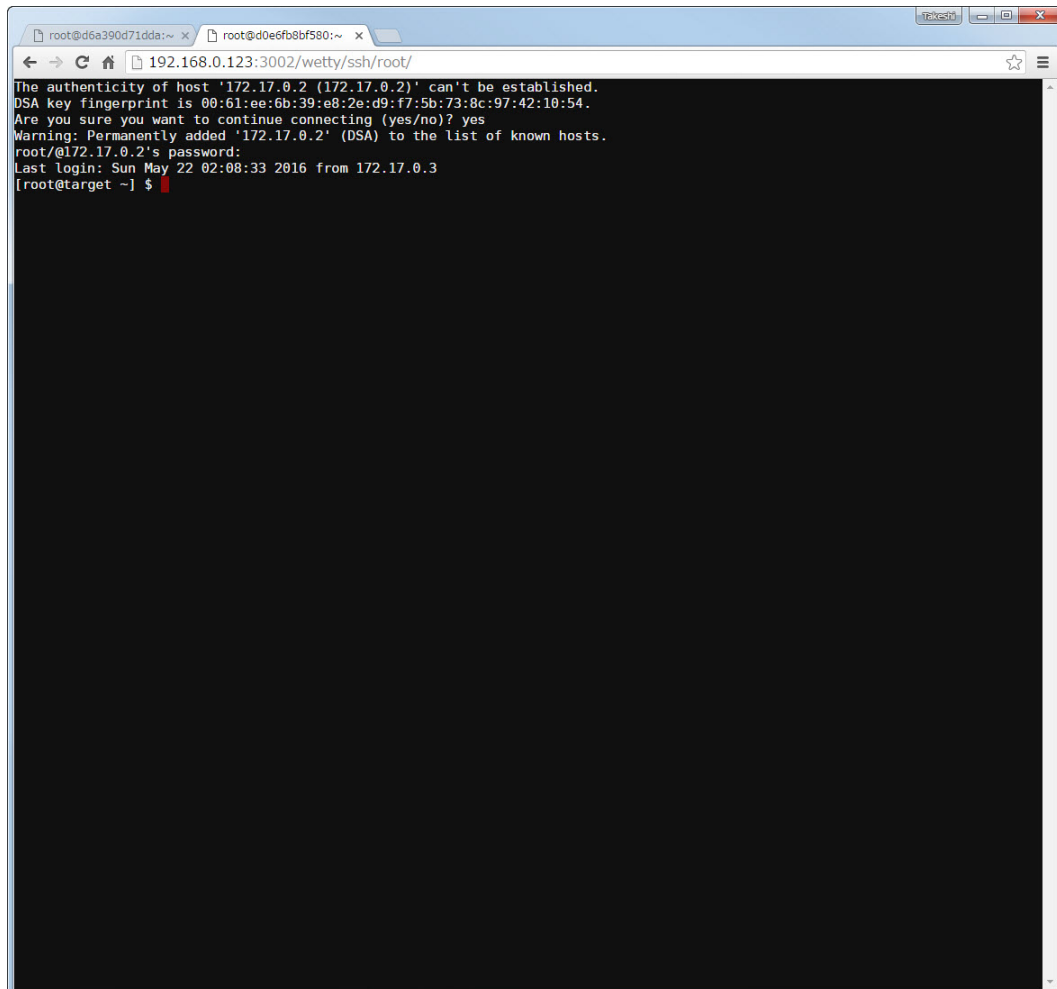
本ハンズオンを通じて Serverspec の簡単な利用方法を体験することができます。ハンズオンでは、2つのマシン Serverspec Host、Serverspec Target を利用します。まず、Target マシンを何も無い状態から、Ansible により Wordpress の初期セットアップが完了した状態にします。その前後で Serverspec を使ってマシンの状態をテストしその結果を比較を通じて利用方法の理解いただく事をゴールとします。

3 ハンズオン実施

3.1 Serverspec Host と Serverspec Target にログイン

Web ブラウザを通じて Serverspec Host と、Serverspec Target にログインします。SSH でログインする際と同じようなコンソール画面が出てきます。初回は yes と入力した後にパスワードを入力してログインします。パスワードはデフォルトで「root」です。なお、Serverspec Host にログインするとプロンプトは「root@host ~] \$」、Serverspec Target にログインするとプロンプトは「root@target ~] \$」になります。

【ログイン画面のサンプル】



【サンプルの入力部分拡大】

```
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
DSA key fingerprint is 00:61:ee:6b:39:e8:2e:d9:f7:5b:73:8c:97:42:10:54.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.17.0.2' (DSA) to the list of known hosts.
root@172.17.0.2's password:
Last login: Sun May 22 02:08:33 2016 from 172.17.0.3
[root@target ~] $
```

Notes

誤ってログアウトしてしまった場合、ブラウザの更新を押すと再度ログインプロンプトに復帰します。

3.2 Serverspec Target にログイン

Serverspec Target マシンにログインします。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。

3.3 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 ディレクトリ内が一覧されれば 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。

3.4 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。

```
[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] $
```


3.5 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

3.6 Serverspec テストの実行（Ansible による自動設定前）

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
```

3.7 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] $
```

3.8 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）となっていることを確認してください。

3.9 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>:<ポート番号>/wordpress>

正しくアクセスできていれば。完了です。

4 参考

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