

Jak uzupełnić automatyczną instalację wewnątrz maszyny wirtualnej - Ansible oraz magia testów infrastruktury z użyciem Chef InSpec

20th November 2019 - Dawid Dacewicz

About me



- ▶ DevOps Engineer
- ▶ Red Hat and AWS certificated
- ▶ Security and DevOps methodology

Contact me:

- ▶ <https://www.linkedin.com/in/dawiddacewicz/>
- ▶ daca444@gmail.com

Terraform integration

► From terraform:

```
provisioner "local-exec" {  
  # method 1: construct inventory from terraform state  
  command = "sleep 60; ansible-playbook -i '${terraform-ansible.public_ip},' server.yml"  
  
  # method 2: use terraform-inventory dynamic inventory script  
  # command = "sleep 90; ansible-playbook -i /usr/local/bin/terraform-inventory  
  server.yml" }
```

► From asible:

- terraform:

```
project_path: '{{ project_dir }}'  
state: present
```

What we want to achieve



Chef Inspec - installation

► Chef Inspec - <https://downloads.chef.io/inspec>

Chef InSpec 4.18.39

Stable Release | [Current Release](#)

Recent releases of our software distributions are under a new license. Our [supported versions](#) page lists which product versions are governed by the new license. Additional information about our licenses can be found [here](#).

Chef InSpec is an infrastructure security and compliance testing framework with a human- and machine-readable language for comparing actual versus desired system state.

JUMP TO OS: [Red Hat Enterprise Linux](#) | [macOS](#) | [SUSE Linux Enterprise Server](#) | [Ubuntu](#) | [Windows](#)

PREVIOUS VERSIONS (STABLE)

4.18.39
4.18.38
4.18.24
4.18.0
4.17.17
4.17.15
4.17.14
4.17.11
4.17.7

Red Hat Enterprise Linux

Red Hat Enterprise Linux 8

[License Information](#)

Architecture: **x86_64**

SHA256: 17c88e3b50d1b221a0731a608a4f40ad24eed89e2fc9fa8d3330e2ebc244abc9

URL: https://packages.chef.io/files/stable/inspec/4.18.39/el/8/inspec-4.18.39-1.el7.x86_64.rpm

Download

Red Hat Enterprise Linux 7

[License Information](#)

Architecture: **x86_64**

SHA256: 17c88e3b50d1b221a0731a608a4f40ad24eed89e2fc9fa8d3330e2ebc244abc9

URL: https://packages.chef.io/files/stable/inspec/4.18.39/el/7/inspec-4.18.39-1.el7.x86_64.rpm

Download

Red Hat Enterprise Linux 6



Inspec shell

- ▶ `inspec shell #Interactive shell`
- ▶ `inspec shell -t ssh://root@1.1.1.1 --password T@jn3! [--sudo]`
- ▶ `inspec shell -t winrm://Administrator@1.1.1.1 --password T@jn3!`
- ▶ `inspec shell -t docker://ee39f68eb241`

```
[root@S2 ~]# inspec shell
Welcome to the interactive InSpec Shell
To find out how to use it, type:
```

```
You are currently running on:
```

```
Name:
Families:
Release:
Arch:
```

```
inspec> package('vim').installed?
=> false
inspec> package('ansible').installed?
=> true
inspec> 
```



Chef Inspec - profile

```
[root@S2 meetup2-master]# inspec init profile httpd
```

InSpec Code Generator

```
Creating new profile at /root/meetup2-master/httpd
```

- Creating file `README.md`
- Creating directory `controls`
- Creating file `controls/example.rb`
- Creating file `inspec.yml`
- Creating directory `libraries`

Chef Inspec - code

```
# copyright: 2018, The Authors
title "Apache httpd,,
control 'httpd-1.0' do
  title 'Verify example website development'
  impact 1.0
  describe package('httpd') do
    it { should be_installed }
  end

  describe service('httpd') do
    it { should be_running }
    it { should be_enabled }
  end

  describe file('/var/www/html/index.html') do
    its('content') { should match /Coming Soon!/ }
  end
end
```


Chef Inspec - check and exec

```
[root@S2 meetup2-master]# inspec exec httpd
```

```
Profile: InSpec Profile (httpd)
```

```
Version: 1.0
```

```
Target: local://
```

- × httpd-1.0: Verify example website development (4 failed)
 - × System Package httpd should be installed
expected that `System Package httpd` is installed
 - × Service httpd should be running
expected that `Service httpd` is running
 - × Service httpd should be enabled
expected that `Service httpd` is enabled
 - × File /var/www/html/index.html content should match /Coming Soon!/
expected nil to match /Coming Soon!/

```
Profile Summary: 0 successful controls, 1 control failure, 0 controls skipped
```

```
Test Summary: 0 successful, 4 failures, 0 skipped
```

Time for ansible

- name: Install apache httpd on RHEL distribution
become: yes
yum:
 - name: "{{ item }}"
 - state: latestloop:
 - httpd
 - unzip
- name: Unarchive deployed package
become: yes
unarchive:
 - src: "{{ WEBSITE_PACKAGE }}"
 - dest: /var/www/html
- name: Apply template
become: yes
template:
 - src: index.html.yml
 - dest: /var/www/html/index.html
- name: check website
uri:
 - url: http://localhost
 - return_content: yes
 - register: resultuntil: result.content.find("Coming Soon!") != -1
retries: 10
delay: 5



Let's run it

```
[root@S2 meetup2-master]# ansible-playbook deploy.yml

PLAY [prod] *****

TASK [Gathering Facts] *****
ok: [s1]

TASK [r_httpd_app : Install apache httpd on RHEL distribution] *****
changed: [s1] => (item=httpd)
ok: [s1] => (item=unzip)

TASK [r_httpd_app : Install apache httpd on Debian distribution] *****
skipping: [s1] => (item=apache2)
skipping: [s1] => (item=unzip)

TASK [r_httpd_app : Start and enable apache on RedHat distribution] *****
changed: [s1]

TASK [r_httpd_app : Start and enable apache on Debian distribution] *****
skipping: [s1]

TASK [r_httpd_app : Unarchive deployed package] *****
changed: [s1]

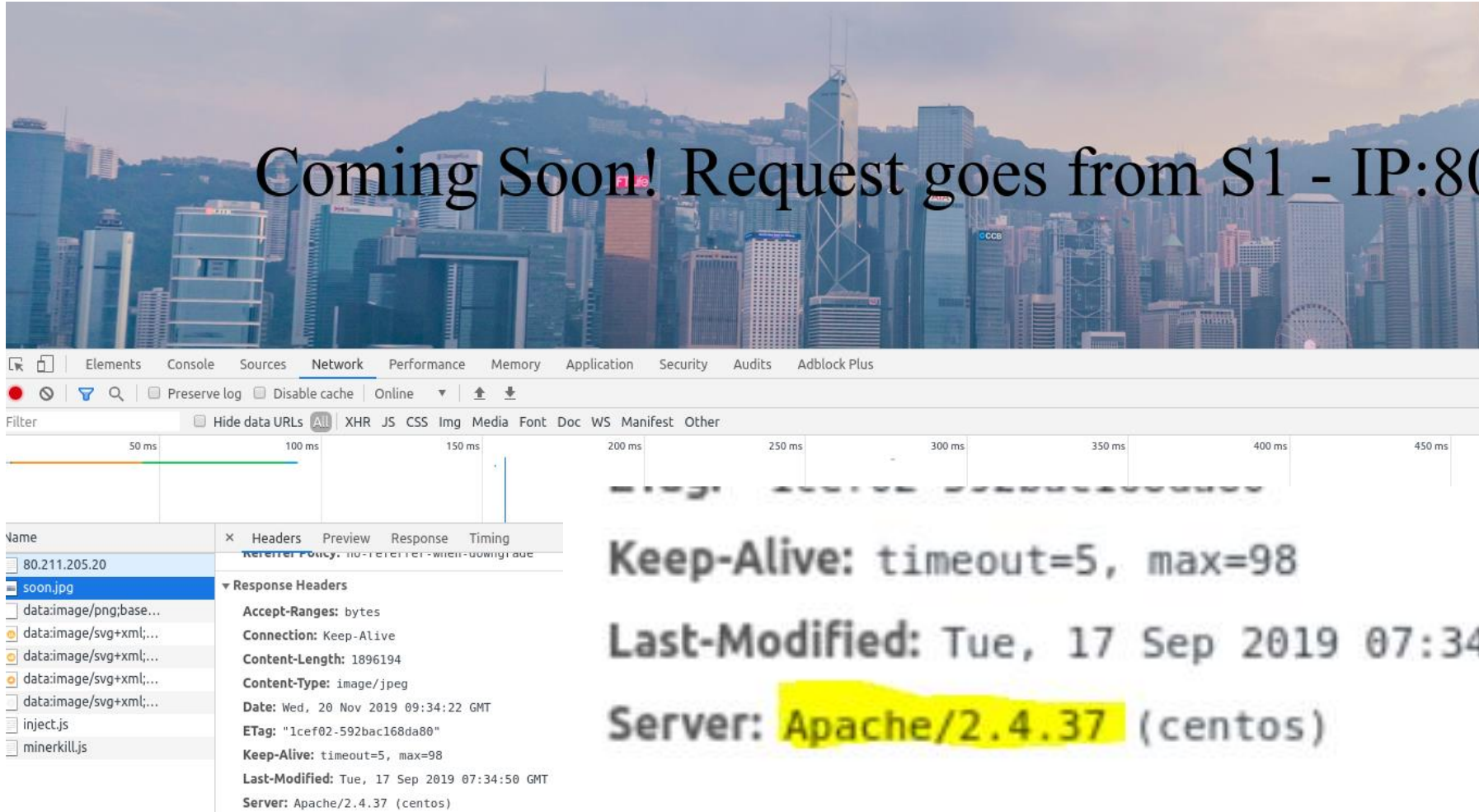
TASK [r_httpd_app : Apply template] *****
changed: [s1]

TASK [r_httpd_app : check website] *****
ok: [s1]

PLAY RECAP *****
s1 : ok=6 changed=4 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
```

What we get

Coming Soon! Request goes from S1 - IP:80



The screenshot displays the Chrome DevTools Network tab. The 'soon.jpg' request is selected, showing its response headers. The headers include:

- Keep-Alive:** timeout=5, max=98
- Last-Modified:** Tue, 17 Sep 2019 07:34:50 GMT
- Server:** Apache/2.4.37 (centos)

The 'Server' header is highlighted in yellow in the original image. The background of the slide features a cityscape of Hong Kong.

Let's set some security

```
control 'httpd-2.0' do
  title 'Check security config'
  impact 0.5
```

```
  describe apache_conf do
    its('ServerTokens') {should cmp 'Prod'}
  end
```

```
  describe apache_conf do
    its('ServerSignature') {should cmp 'Off'}
  end
end
```

```
# copyright: 2018, The Authors

title "Apache httpd"

[control 'httpd-1.0' do
  title 'Verify example website development'
  impact 1.0

  describe package('httpd') do
    it { should be_installed }
  end

  describe service('httpd') do
    it { should be_running }
    it { should be_enabled }
  end

  describe file('/var/www/html/index.html') do
    its('content') { should match /Coming Soon!/ }
  end
end

[control 'httpd-2.0' do
  title 'Check security config'
  impact 0.5

  describe apache_conf do
    its('ServerTokens') {should cmp 'Prod'}
  end

  describe apache_conf do
    its('ServerSignature') {should cmp 'Off'}
  end
end]
```



Chef inspec again

```
[root@S2 meetup2-master]# inspec exec httpd -t ssh://80.211.205.20 -i ~/.ssh/id_rsa
```

```
Profile: InSpec Profile (httpd)
```

```
Version: 1.0
```

```
Target:  ssh://root@80.211.205.20:22
```

```
✓ httpd-1.0: Verify example website development
  ✓ System Package httpd should be installed
  ✓ Service httpd should be running
  ✓ Service httpd should be enabled
  ✓ File /var/www/html/index.html content should match /Coming Soon!/
× httpd-2.0: Check security config (2 failed)
  × Apache Config /etc/httpd/conf/httpd.conf ServerTokens should cmp == "Prod"

    expected: "Prod"
    got: nil

    (compared using `cmp` matcher)

  × Apache Config /etc/httpd/conf/httpd.conf ServerSignature should cmp == "Off"

    expected: "Off"
    got: nil

    (compared using `cmp` matcher)
```

```
Profile Summary: 1 successful control, 1 control failure, 0 controls skipped
```

```
Test Summary: 4 successful, 2 failures, 0 skipped
```

Let's fix ansible then

- name: Set ServerTokens Prod
lineinfile:
 path: /etc/httpd/conf/httpd.conf
 regexp: '^ServerTokens'
 line: 'ServerTokens Prod'
 backup: yes

- name: Set ServerSignature Off
lineinfile:
 path: /etc/httpd/conf/httpd.conf
 regexp: '^ServerSignature'
 line: 'ServerSignature Off'
 backup: yes

- name: Restart apache httpd
service:
 name: httpd
 state: restarted

Run it again

```
TASK [r_httpd_app : Apply template] *****
ok: [s1]

TASK [r_httpd_app : check website] *****
ok: [s1]

TASK [r_httpd_app : Set ServerTokens Prod] *****
changed: [s1]

TASK [r_httpd_app : Set ServerSignature Off] *****
changed: [s1]

TASK [r_httpd_app : Restart apache httpd (shoul be as handler - just for easy readable)] *****
changed: [s1]

PLAY RECAP *****
s1                : ok=9    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```


Run it again

```
[root@S2 meetup2-master]# inspec exec httpd -t ssh://80.211.205.20 -i ~/.ssh/id_rsa
```

Profile: InSpec Profile ([httpd](#))

Version: 1.0

```
Target:  ssh://root@80.211.205.20:22
```

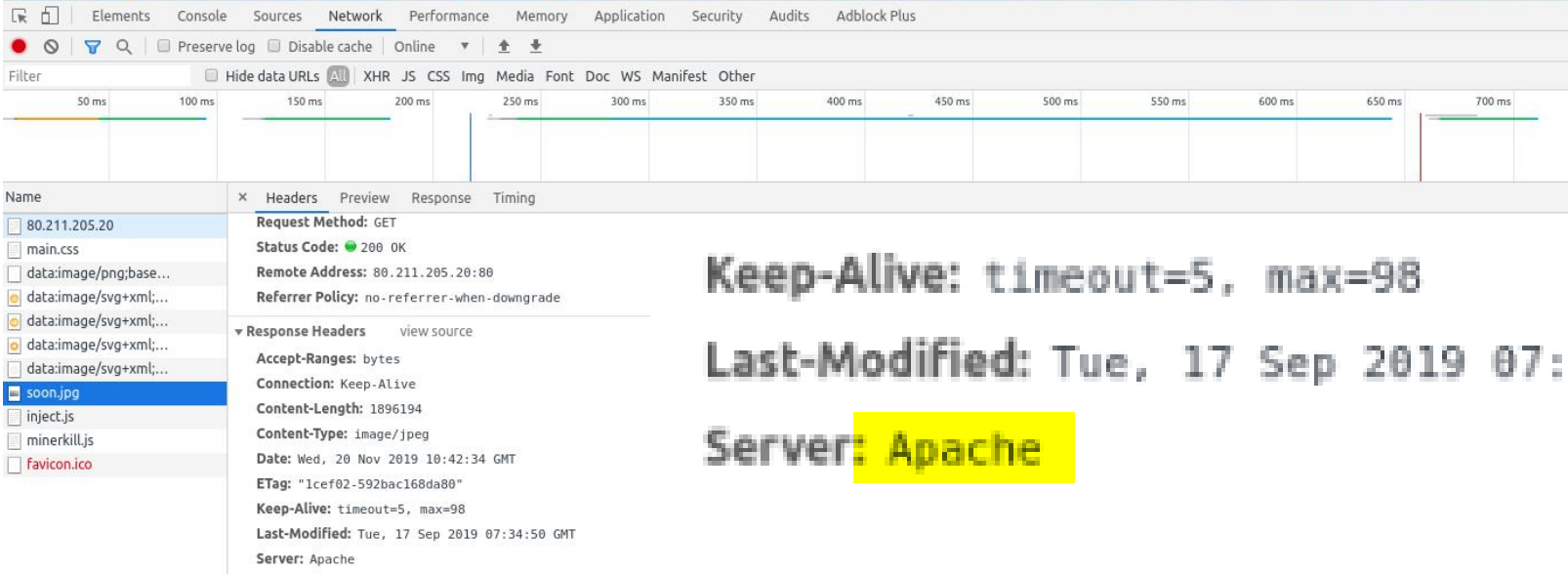
- ✓ httpd-1.0: Verify example website development
 - ✓ System Package httpd should be installed
 - ✓ Service httpd should be running
 - ✓ Service httpd should be enabled
 - ✓ File /var/www/html/index.html content should match /Coming Soon!/
 - ✓ Content type should be application/html
- ✓ httpd-2.0: Check security config
 - ✓ Apache Config /etc/httpd/conf/httpd.conf ServerTokens should cmp == "Prod"
 - ✓ Apache Config /etc/httpd/conf/httpd.conf ServerSignature should cmp == "Off"

```
Profile Summary: 2 successful controls, 0 control failures, 0 controls skipped
```

Test Summary: 6 successful, 0 failures, 0 skipped



Results



Keep-Alive: timeout=5, max=98

Last-Modified: Tue, 17 Sep 2019 07:34:50 GMT

Server: Apache



Q & A