INFRASTRUCTURE DEVELOPMENT USING CUCUMBER, SEVERSPEC & ANSIBLE

ThoughtWorks®

Itamar Hassin June 2015

THE GOAL

- Communicate across silos
- Optimise SA's time (reduce manual VDDs)
- Avoid snowflakes
- Fail fast (offline!) and cycle quickly
- Find about local/remote environment differences
- **Time To Rebuild < Time To Fix**

IAC, BDD, MDD

- Infrastructure As Code
- Behaviour-Driven Development
- Monitor-Driven Development

INFRASTRUCTURE AS CODE

A way by which we can define, provision, and configure hardware and Virtual Machines to suite our needs

BEHAVIOUR DRIVEN DEVELOPMENT

An activity whereby a feature's behaviour is described *prior* to its implementation

MONITOR DRIVEN DEVELOPMENT

- Write production monitors based on desired result
- Implement enough infrastructure to satisfy the monitor

OUR TOOLBOX

TDD	Provisioning	BDD	IAC
ServerSpec	Vagrant	Cucumber	Ansible

SERVERSPEC

With Serverspec, you can write tests for checking that servers are configured correctly.

VAGRANT

Vagrant is computer software for creating and configuring virtual development environments.

CUCUMBER

Simple, human collaboration

ANSIBLE

Ansible is a radically simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.

SALLY & LEO AT WORK

Danielle:	I want a web server.
Leo:	Sure, boss.
Danielle:	We’re a silo-ed org.
Sally:	Sure, boss.
Danielle:	The Admins need specs.
Sally:	Sure, boss.
Leo:	Sally, let’s pair.

WORKFLOW

How Sally and Leo go about setting up a web server in such a way that it

- Is automated/repeatable
- Assures environments to be consistent for development & testing & deploying

SALLY WRITES SPECS (IN A WORD DOC)

```
Feature:
As a Webmaster (yeah baby!) I need a webserver to be running
so that I may master it.

Background:
Given my server is available on "33.33.33.22"
When I provision it

Scenario:
When I get access to it
Then I expect it to have apache running
```

THEY USE BDD

```
$ cucumber
No such file or directory @ rb_sysopen - features. Please
create a features directory to get started. (Errno::ENOENT)
```

MAKE FEATURES DIRECTORY

```
$ mkdir features
```

CUCUMBER IS SET UP

```
$ cucumber
0 scenarios
0 steps
0m0.000s
```

PASTE FEATURE (FROM WORD DOC)

```
Feature:
As a Webmaster (yeah baby!) I need a webserver to be running
so that I may master it.

Background:
Given my server is available on ”33.33.33.22”
When I provision it

Scenario:
When I get access to it
Then I expect it to have apache running
```

MISSING STEPS

```
$ cucumber
Feature:
  As a Webmaster (yeah baby!) I need a webserver to be running so that I may master it.

  Background:                    # features/httpd.feature:4
    Given my server is available # features/httpd.feature:5
    And I provision it           # features/httpd.feature:6

  Scenario:                                         # features/httpd.feature:8
    When I get access to it                         # features/httpd.feature:9
    Then I expect it to have apache running # features/httpd.feature:10

1 scenario (1 undefined)
4 steps (4 undefined)
0m0.002s

You can implement step definitions for undefined steps with these snippets:

Given(/^my server is available on "(.*?)"$/) do |arg1|
  pending # express the regexp above with the code you wish you had
end

When(/^I provision it$/) do
  pending # express the regexp above with the code you wish you had
end
```

WORKFLOW

At this stage, the high-level functional requirement needs more specification, so they unit test the server

THEY WRITE A SPEC

```
Test files must be placed under the directory
which name matches the target host name:

spec/target.example.com/http_spec.rb
```

THEY WRITE THE EXPECTATION

```
describe package('apache2') do
  it { should be_installed }
end

describe service('apache2') do
  it { should be_enabled   }
  it { should be_running   }
end

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

describe file('/etc/apache2/sites-enabled/000-default.conf') do
  it { should be_file }
  it { should contain "example.com" }
end
```

MISSING VAGRANT

```
$ rake spec

A Vagrant environment or target machine is required
to run this
command.

Run `vagrant init` to create a new Vagrant
environment.

Or, get an ID of a target machine from `vagrant
global-status` to run
this command on.
```

THEY CREATE A VAGRANTFILE

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "Ubuntu 14.04"
  config.vm.define "webserver"
  config.vm.hostname = "webserver"

 config.vm.network "private_network", ip: "33.33.33.22"

  config.vm.provider "virtualbox" do |vb|
    vb.customize ["modifyvm", :id, "--memory", "1024"]
  end
end
```

THEY RUN MONITOR AGAIN

```
$ rake spec
ruby -S rspec spec/webserver/httpd_spec.rb
FFFFFF

Failures:

  1) Package "apache2" should be installed
     Failure/Error: it { should be_installed }
       sudo dpkg-query -f '${Status}' -W apache2 | grep -E '^(install|hold)
ok installed$'
       expected Package "apache2" to be installed
     # ./spec/webserver/httpd_spec.rb:4:in `block (2 levels) in <top
(required)>'

  2) Service "apache2" should be enabled
     Failure/Error: it { should be_enabled   }
       sudo ls /etc/rc3.d/ | grep -- '^S..apache2' || sudo grep 'start
on' /etc/init/apache2.conf
       expected Service "apache2" to be enabled
     # ./spec/webserver/httpd_spec.rb:8:in `block (2 levels) in <top
(required)>'
```

THEY PROVISION A VM

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "Ubuntu 14.04"
  config.vm.define "webserver"
  config.vm.hostname = "webserver"

  config.vm.network "private_network", ip: "33.33.33.22"

  config.vm.provider "virtualbox" do |vb|
    vb.customize ["modifyvm", :id, "--memory", "1024"]
  end

  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "playbook.yml"
    ansible.inventory_path = "inventory.ini"
    ansible.sudo = true
  end
end
```

THEY RUN THE UNIT TESTS

```
$ rake spec
ruby -S rspec spec/webserver/httpd_spec.rb
There are errors in the configuration of this machine.
Please fix
the following errors and try again:

ansible provisioner:
* `playbook` for the Ansible provisioner does not exist on
the host system: playbook.yml


......

Finished in 3.51 seconds
```

WRITE THE PLAYBOOK

```yaml
---
- hosts: all
  user: vagrant
  sudo: true

  roles:
    - common
    - apache
```

COMMON ROLE EXAMPLE

```
 - name: Update apt cache if needed
   apt: update_cache=yes cache_valid_time=3600

 - name: Upgrade OS
   apt: upgrade=dist force=yes

 - name: Install needed packages
   apt: pkg={{item}} state=installed
   with_items:
     - cron
     - logrotate
     - curl
     - git
     - update-motd

 - name: Create the deploy user
   user: name={{user}} comment="deploy user" generate_ssh_key=yes
ssh_key_bits=2048 state=present password={{password}} shell=/bin/bash

 - name: Set {{user}} as sudoer
   lineinfile: dest=/etc/sudoers line="{{user}} ALL=(ALL) NOPASSWD ":" ALL"

 - name: remove ubuntu's user
   user: name=ubuntu state=absent remove=yes
```

THEY WRITE INVENTORY

```
[webserver]
33.33.33.22
```

THEY RUN VAGRANT

```
$ vagrant provision webserver
==> webserver: Running provisioner: ansible...

PLAY [webserver] **************************************************************

GATHERING FACTS ***************************************************************
ok: [webserver]

TASK: [kamaln7.swapfile | Create swapfile] ************************************
changed: [webserver]

TASK: [common | Update apt cache if needed] ***********************************
ok: [webserver]

TASK: [common | Upgrade OS] ***************************************************
ok: [webserver]

TASK: [common | Install needed packages] **************************************
ok: [webserver] => (item=cron,logrotate,curl,git,update-motd)

TASK: [common | Create the deploy user] ***************************************
ok: [webserver]
```

WORKFLOW

At this stage, Vagrant and Ansible scripts provision and configure the server; now tie that to the suspended Cucumber spec

STEP IMPLEMENTATION FOR CUCUMBER

```
Given(/^my server is available on "(.*?)"$/) do |ip_address|
 @ip_address = ip_address
 output=`vagrant up`
end

And(/^I provision it$/) do
 output=`vagrant provision`
end

When(/^I get access to it$/) do
  run_remote("ls")
end

Then(/^I expect it to have apache running$/) do
  run_remote("ps asx | grep apache")
end

def run_remote(command)
  Net::SSH.start(@ip_address, "vagrant", :password => "vagrant") do |ssh|
    result = ssh.exec!(command)
  end
end
```

LEO RUNS THE UNIT TESTS

```
$ rake spec
ruby -S rspec spec/webserver/httpd_spec.rb
......

Finished in 3.73 seconds
6 examples, 0 failures
```

SALLY RUNS THE SPECS

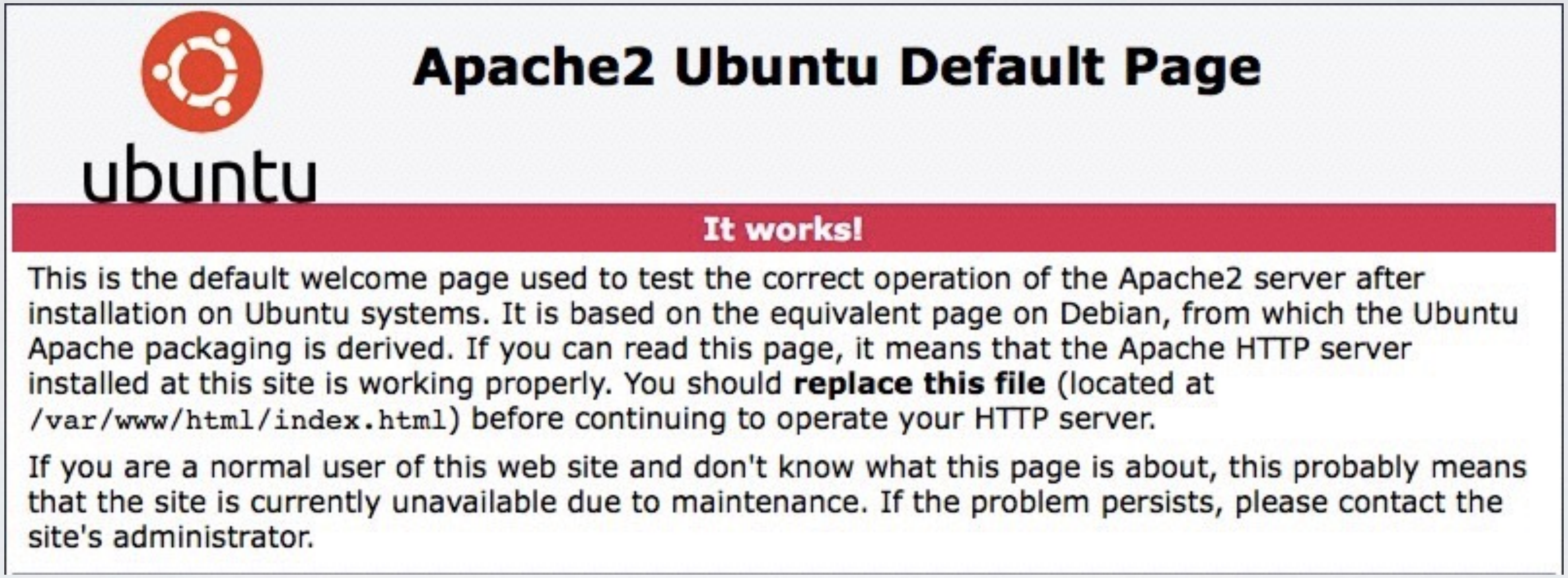
```
$ cucumber
Feature:
  As a Webmaster (yeah baby!) I need a webserver to be running so
that I may master it.

  Background:                    # features/httpd.feature:4
    Given my server is available # features/steps/httpd_steps.rb:3
    And I provision it           # features/steps/httpd_steps.rb:8

  Scenario:                     # features/httpd.feature:8
    When I get access to it     # features/steps/httpd_steps.rb:12
    Then I expect it to have apache running

1 scenario (1 passed)
4 steps (4 passed)
0m8.433s
```

DANIELLE SEES APACHE RUN



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 is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

SOURCE CONTROL

SCaaCC

- Infrastructure code is part of the app code
- Developers and SysAdmins share code using git for modifications.
- HugOps use **master** (others can use **forking**)

AUTOMATION

- Commit the Cucumber and ServerSpec source code to git
- Add a Jenkins task to run the scripts as part of the integration test suite
- Add ServerSpec as a monitor in dev to guard against configuration regression

WORK AS A SINGLE TEAM

- SysAdmins are part of the development team
- Participate in standups, part of a story/task cards
- Participate in pairing sessions
- Empathise

REFERENCES

The Visible Ops Handbook

James White's manifesto

Ansible

Cucumber

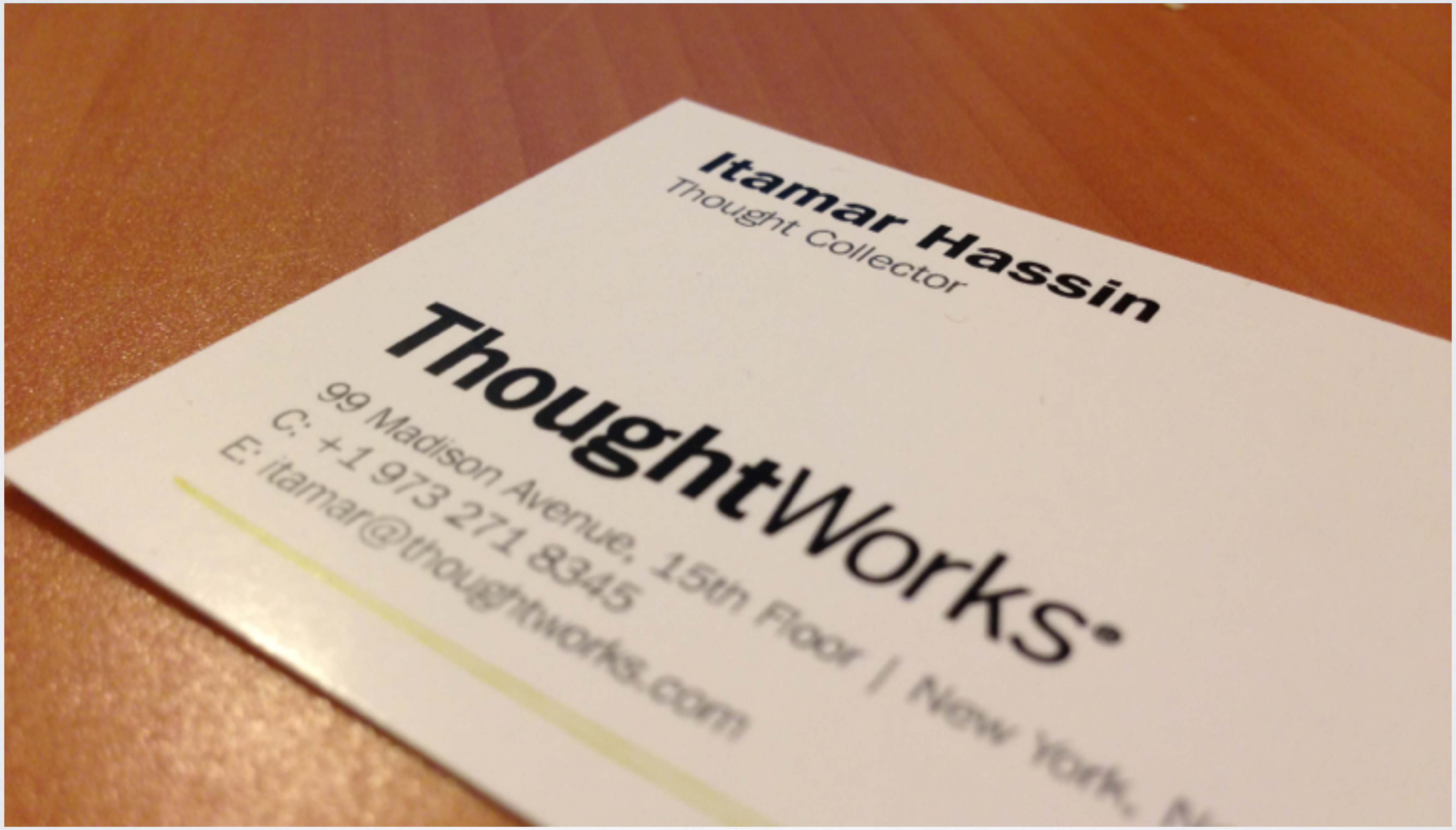
ServerSpec

DevOps Conferences

THANK YOU!

@itababy

www.in-context.com



Itamar Hassin
Thought Collector

ThoughtWorks®

99 Madison Avenue, 15th Floor | New York,
C: +1 973 271 8345
E: itamar@thoughtworks.com