

Lab 7 Setup Windows Chef with Vagrant

Nonetheless, the selection of “Boxes” (VM templates) on vagrantcloud.com is pretty limited right now, presumably due to licensing concerns.

Search boxes

Box names, descriptions, users and providers can be used to find what you're looking for.

Filter boxes to contain provider:

virtualbox

vmware_desktop

digitalocean

aws

rackspace

hyperv


parallels

Sort by:

Recently Created


Recently Updated

Favorites

 [opentable/win-2012r2-standard-amd64-nocm](#)

Version 0.2.0 ★ 22 favorites

Windows 2012 R2 Standard Edition x64 (with no configuration management tools)

 [kensykora/windows_2012_r2_standard](#)

Version 0.5.0 ★ 14 favorites

Windows Server 2012 R2 Standard

The most popular Windows 2012 R2 box is currently one provided by [OpenTable](#), however it seems to have issues with password expiry, so, we'll go with the *second* most popular, the one by [kensykora](#).

If you open up the [link to that box](#), you'll see a handy command in a textbox, ready for you to copy out.

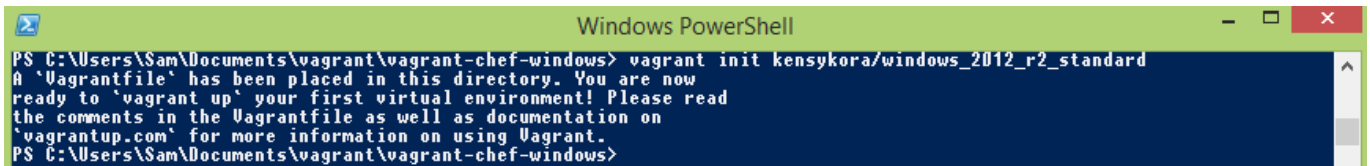
 kensykora / windows_2012_r2_standard

Windows Server 2012 R2 Standard

```
$ vagrant init kensykora/windows_2012_r2_standard
```

Copy that command, open a new PowerShell window on your computer, create a new folder in your My Documents called “vagrant-chef-windows”, then execute the command:

```
vagrant init kensykora/windows_2012_r2_standard
```



```
Windows PowerShell
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows> vagrant init kensykora/windows_2012_r2_standard
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows>
```

This creates a *Vagrantfile* in the directory in which you’ve executed the command.

1) Setup Initial Vagrant Configuration

Open the Vagrantfile in your favorite text editor, and replace the contents with the following:

```
VAGRANTFILE_API_VERSION = '2'
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  # Every Vagrant virtual environment requires a box to build off of.
  config.vm.box = 'kensykora/windows_2012_r2_standard'

  # Forward ports
  config.vm.network 'forwarded_port', guest: 80, host: 8080

  config.vm.provider 'virtualbox' do |vb|
    # Don't boot with headless mode
    vb.gui = true
  end

  # Shell Provisioning
  config.vm.provision 'shell' do |shell|
    shell.path = 'install-chef.ps1'
  end
end
```

The configuration file is Ruby based, and does several things.

1. Provisions the VM based on kensykora/windows2012r2_standard (downloading it if necessary)
2. Forwards port 80 in the guest machine to port 8080 on your machine (the host)
3. Pops up a Virtualbox window with the guest's console for simplicity's sake
4. Executes install-chef.ps1 in the guest

Take a few moments to pair up the list above with the lines in the configuration file, once you have, you'll wonder "where the hell is it getting install-chef.ps1 from?". At the moment, it isn't.

2) Use PowerShell Bootstrapping to Install Chef

Create a new file in your *vagrant-chef-windows* directory called **install-chef.ps1* and populate it with the following:

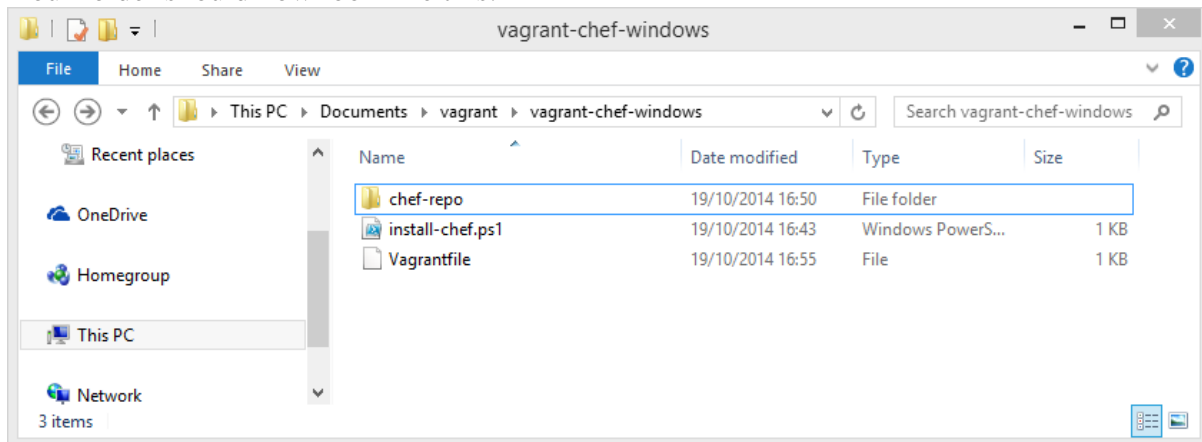
```
$progressPreference = 'silentlyContinue';
$chefInstaller = 'C:vagrantchef-windows-11.16.2-1.windows.msi';
$chefInstallerUri = "https://opscode-omnibus-
packages.s3.amazonaws.com/windows/2008r2/x86_64/chef-windows-11.16.2-
1.windows.msi";

if(!(test-path $chefInstaller)){
    Write-Host "$(Get-Date) Downloading Chef...";
    Invoke-WebRequest -Uri $chefInstallerUri -outfile $chefInstaller;
}

if(!(Test-Path "C:chef")){
    Write-Host " $(Get-Date) Installing Chef";
    Start-Process -Wait -FilePath 'C:\Windows\system32\msiexec.exe' -
ArgumentList @('-i', $chefInstaller, '/quiet', '/log', 'C:\tmp\chef-client-
install.log')
    Write-Host " $(Get-Date) Installation Complete"
}else{
    Write-Host " $(Get-Date) Chef is already installed!";
}
```

Ideally, we wouldn't need to do this as Chef would already be installed in the Box we got from VagrantCloud.com, however, at the time of writing there are no Windows 2012 R2 boxes with Chef pre-installed.

Your folder should now look like this:



3) Power On – Vagrant Up

Now, ensure you're in your *vagrant-chef-windows* folder in the PowerShell console, then execute:

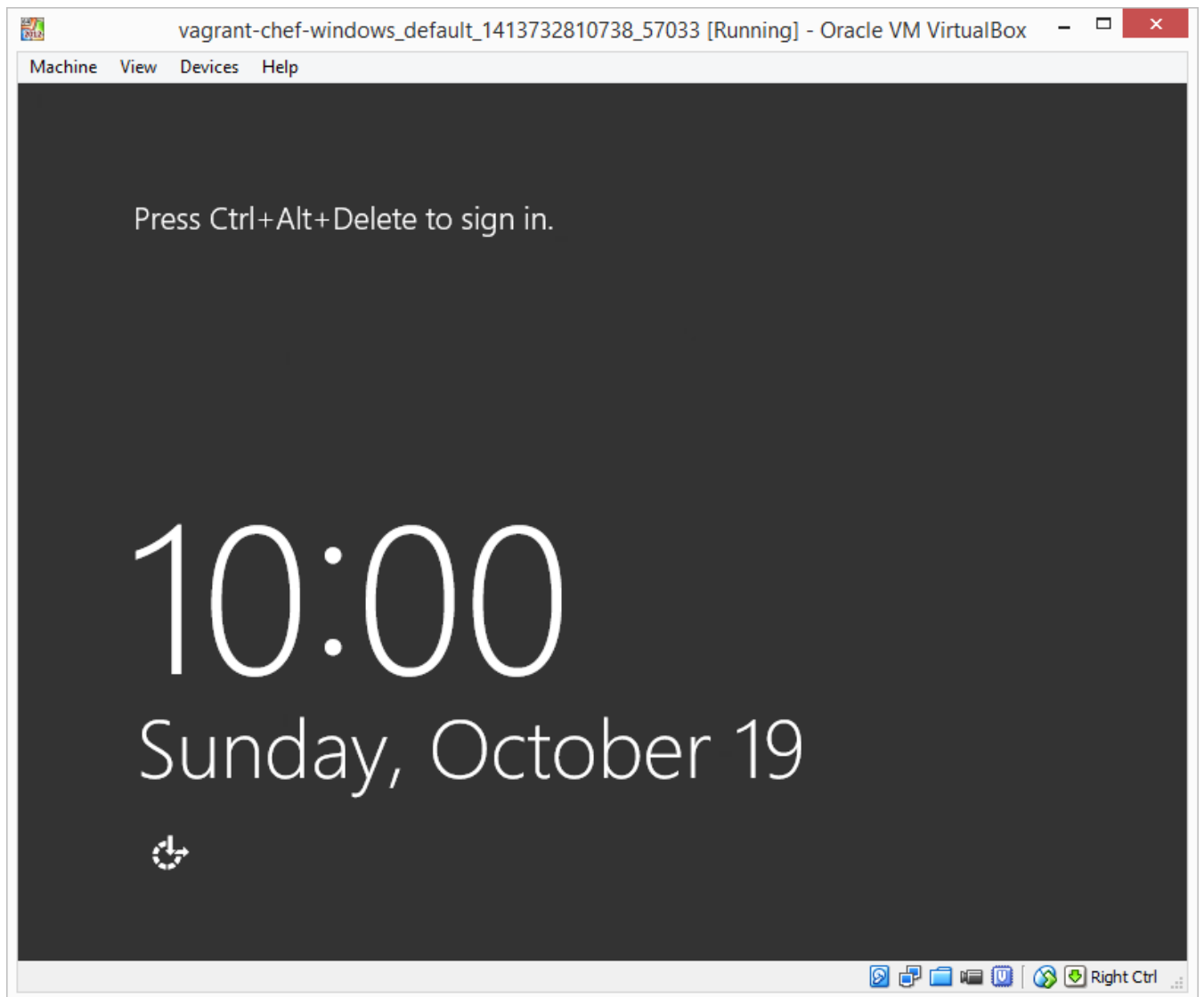
```
vagrant up
```

```
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'kensyhora/windows_2012_r2_standard'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'kensyhora/windows_2012_r2_standard' is up to date...
==> default: Setting the name of the VM: vagrant-chef-windows_default_1413732810738_57033
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
==> default: Forwarding ports...
default: 3389 => 3389 (adapter 1)
default: 80 => 8080 (adapter 1)
default: 22 => 2222 (adapter 1)
default: 5985 => 55985 (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Mounting shared folders...
default: /vagrant => C:/Users/Sam/Documents/vagrant/vagrant-chef-windows
==> default: Running provisioner: shell...
default: Running: c:\tmp\vagrant-shell.ps1
==> default: 10/19/2014 08:35:00 Installing Chef
==> default: 10/19/2014 08:36:05 Installation Complete
```

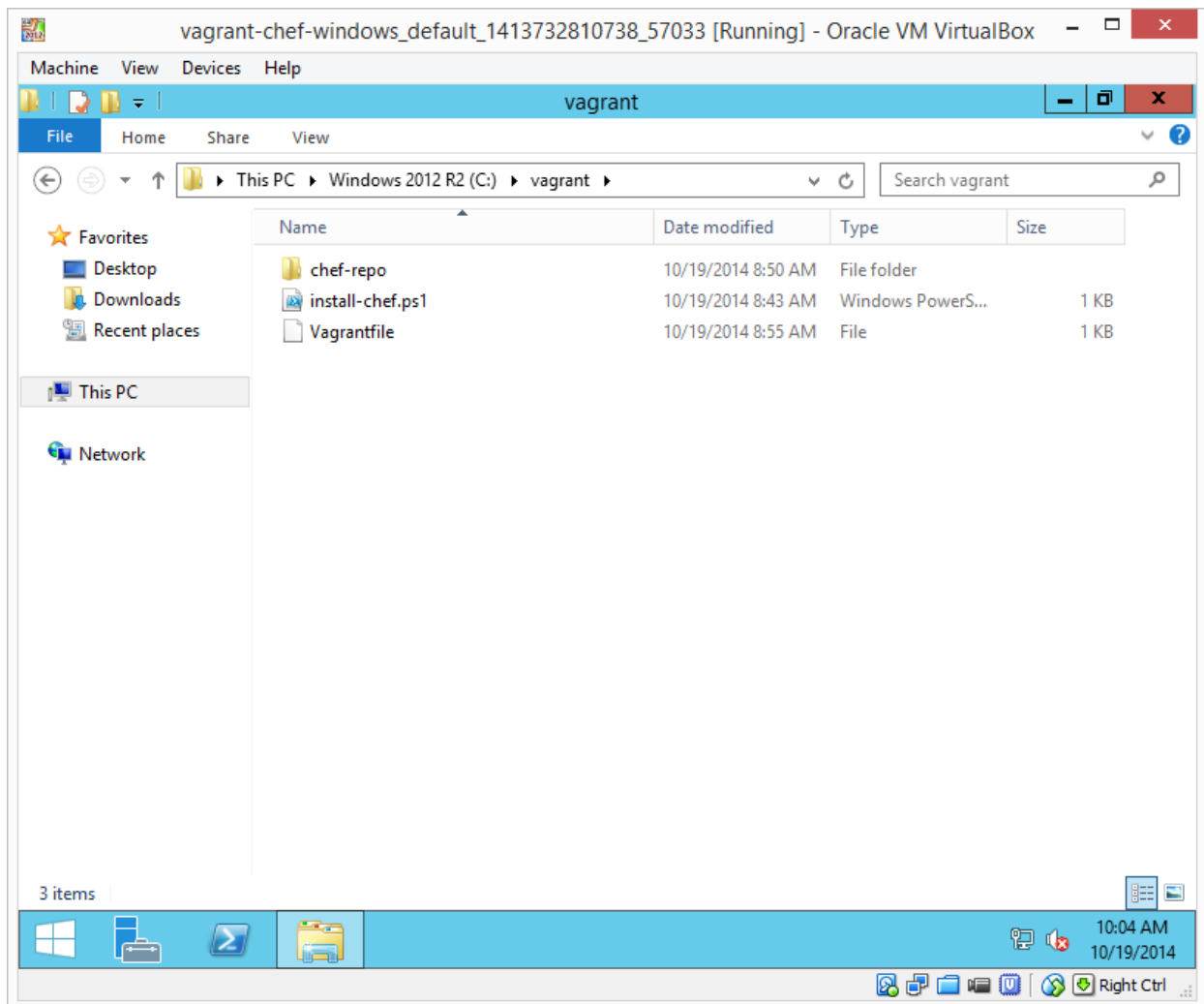
It will scurry off, download the kensyhora 2012 R2 box (not shown as I already had it), power up a new VM and execute your ps1. Once complete, you should have a VirtualBox console pop up and allow you to sign in (right ctrl + del = Ctrl + Alt + Delete).

Username: Vagrant

Password: vagrant



If you login, you'll see *C:\chef* exists, and if you browse into *C:\vagrant*, you'll see that the entirety of your *vagrant-chef-windows* folder is available within the VM!



This is **important** because almost all file paths you'll set in your Vagrantfile configuration will be relative to this directory.

4) Setup Vagrant Chef Provisioning Configuration

Now it's time to actually *use* Chef. But we're not going to just open up a PS console inside the VM and run chef-client. Oh no, we're going to use [Vagrant's chef-client provisioning](#) functionality!

That means that every time we deploy a new VM, our PS1 file will install Chef, then Vagrant will run chef-client for us, with the configuration we've defined in the Vagrantfile.

Add the following lines to the end of your Vagrantfile (but before the final "end").

```
# Chef Provisioning
config.vm.provision 'chef_client' do |chef|
  chef.chef_server_url = 'https://api.opscode.com/organizations/orgname'
  chef.node_name = 'node20141019'
```

```

chef.validation_client_name = 'orgname-validator'
chef.validation_key_path = "chef-repo\\.chef\\orgname-validator.pem"
chef.add_recipe 'learn_chef_iis'
end

```

You will, of course, need to replace *orgname* with your organisation name on the highlighted lines, and amend the node_name if you like.

Your Vagrantfile should now look like this:

```

1  # Vagrantfile API/syntax version. Don't touch unless you know what you're doing!
2  VAGRANTFILE_API_VERSION = "2"
3
4  Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
5      # Every Vagrant virtual environment requires a box to build off of.
6      config.vm.box = "kensyhora/windows_2012_r2_standard"
7
8      # Forward ports
9      config.vm.network "forwarded_port", guest: 80, host: 8080
10
11     config.vm.provider "virtualbox" do |vb|
12         # Don't boot with headless mode
13         vb.gui = true
14     end
15
16     # Shell Provisioning
17     config.vm.provision "shell" do |shell|
18         shell.path = "install-chef.ps1"
19     end
20
21     # Chef Provisioning
22     config.vm.provision "chef_client" do |chef|
23         chef.chef_server_url = "https://api.opscode.com/organizations/samartin"
24         chef.node_name = "node20141019"
25         chef.validation_client_name = "samartin-validator"
26         chef.validation_key_path = "chef-repo\\.chef\\samartin-validator.pem"
27         chef.add_recipe "learn_chef_iis"
28     end
29
30 end
31

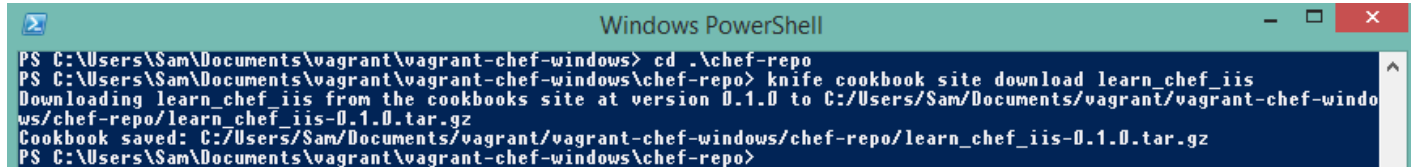
```

This code uses the Chef Client we've already installed and the *orgname-validator.pem* which came with our Starter Kit in order to add this guest as a node to our managed Chef environment.

5) Upload the Cookbook

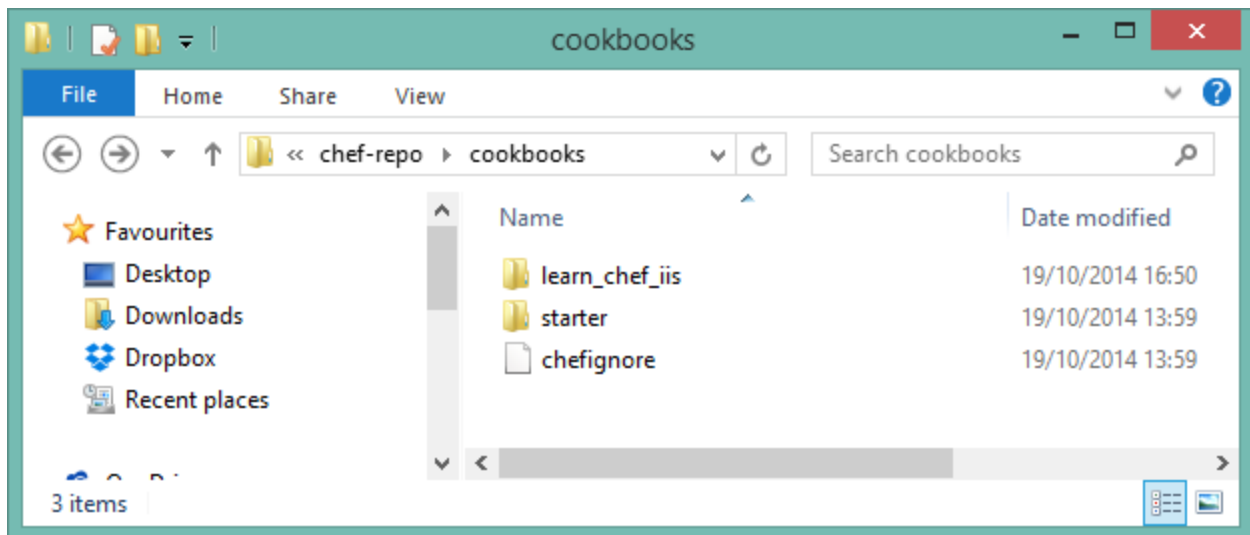
But wait, we haven't got the cookbook *learn_chef_iis* (a simple Windows/IIS example used by the learnchef.com/windows walkthroughs)! CD into your *chef-repo* directory and execute:

```
knife cookbook site download learn_chef_iis
```



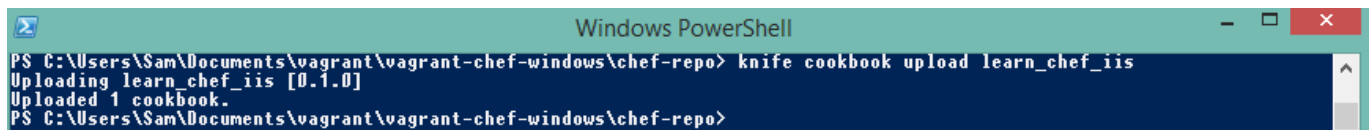
```
Windows PowerShell
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows> cd .\chef-repo
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows\chef-repo> knife cookbook site download learn_chef_iis
Downloading learn_chef_iis from the cookbooks site at version 0.1.0 to C:/Users/Sam/Documents/vagrant/vagrant-chef-windows/chef-repo/learn_chef_iis-0.1.0.tar.gz
Cookbook saved: C:/Users/Sam/Documents/vagrant/vagrant-chef-windows/chef-repo/learn_chef_iis-0.1.0.tar.gz
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows\chef-repo>
```

Now extract the resulting tar.gz into your cookbooks subdir.



And finally, upload it to your managed Chef environment.

```
knife upload learn_chef_iis
```



```
Windows PowerShell
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows\chef-repo> knife cookbook upload learn_chef_iis
Uploading learn_chef_iis [0.1.0]
Uploaded 1 cookbook.
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows\chef-repo>
```

6) Vagrant Provision

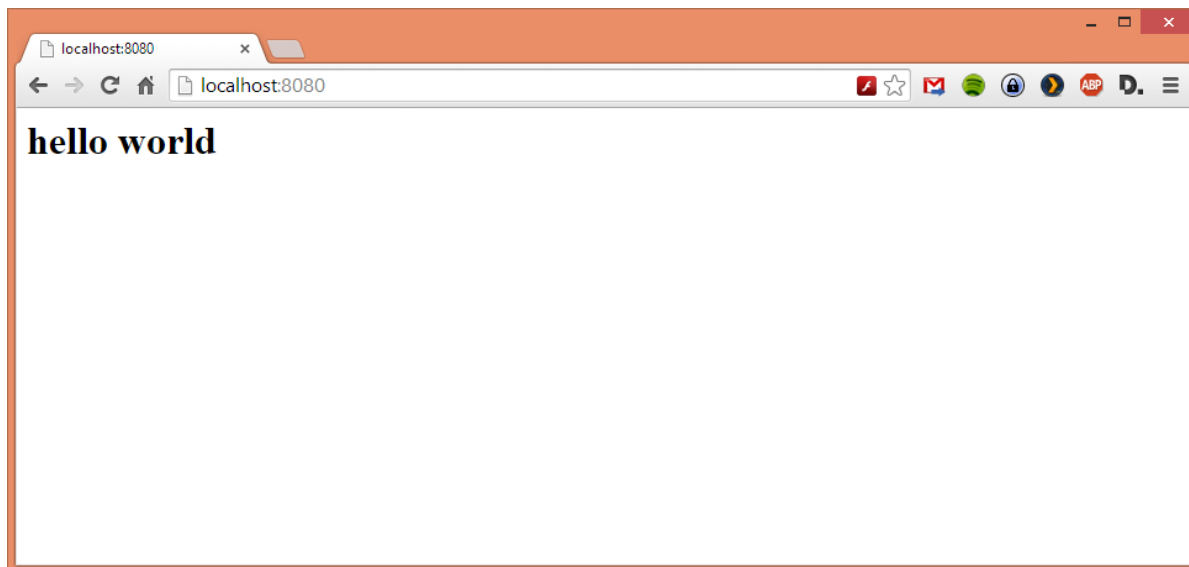
Excellent! The cookbook's ready to go. Now CD up a level into your vagrant directory and run:

```
vagrant provision
```



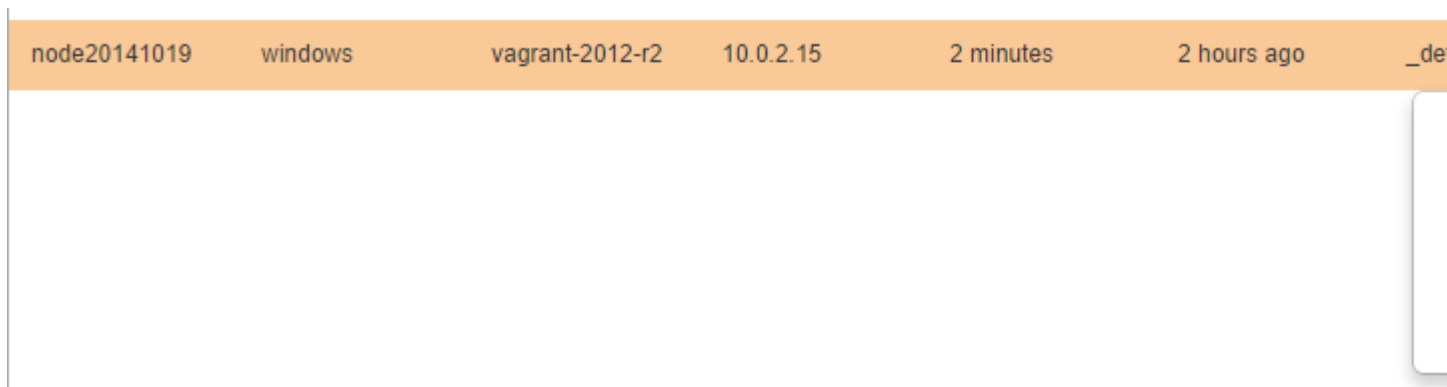
```
Windows PowerShell
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows> vagrant provision
==> default: Running provisioner: shell...
==> default: Running: c:\tmp\vagrant-shell.ps1
==> default: 10/19/2014 10:40:26 Chef is already installed!
==> default: Running provisioner: chef_client...
==> default: Creating folder to hold client key...
==> default: Uploading chef client validation key...
Generating chef JSON and uploading...
==> default: Running chef-client...
==> default: [2014-10-19T10:40:35-07:00] WARN:
==> default: *****
==> default: SSL validation of HTTPS requests is disabled. HTTPS connections are still
==> default: encrypted, but chef is not able to detect forged replies or man in the middle
==> default: attacks.
==> default:
==> default: To fix this issue add an entry like this to your configuration file:
==> default:
==> default: ```
==> default: # Verify all HTTPS connections (recommended)
==> default:   ssl_verify_mode :verify_peer
==> default:
==> default: # OR, Verify only connections to chef-server
==> default:   verify_api_cert true
==> default: ```
==> default:
==> default: To check your SSL configuration, or troubleshoot errors, you can use the
==> default: 'knife ssl check' command like so:
==> default:
==> default: ```
==> default: knife ssl check -c c:\tmp\vagrant-chef-4\client.rb
==> default: ```
==> default:
==> default: *****
==> default: [2014-10-19T10:40:38-07:00] INFO: *** Chef 11.16.2 ***
==> default:
==> default: [2014-10-19T10:40:38-07:00] INFO: Chef-client pid: 2488
==> default: [2014-10-19T10:41:06-07:00] INFO: Client key /etc/chef/client.pem is not present - registering
==> default: [2014-10-19T10:41:08-07:00] INFO: HTTP Request Returned 404 Object Not Found: error
==> default: [2014-10-19T10:41:09-07:00] INFO: Setting the run_list to ["recipe[learn_chef_iis]"] from CLI options
==> default: [2014-10-19T10:41:09-07:00] INFO: Run List is [recipe[learn_chef_iis]]
==> default: [2014-10-19T10:41:09-07:00] INFO: Run List expands to [learn_chef_iis]
==> default: [2014-10-19T10:41:09-07:00] INFO: Starting Chef Run for node20141019
==> default: [2014-10-19T10:41:09-07:00] INFO: Running start handlers
==> default: [2014-10-19T10:41:09-07:00] INFO: Start handlers complete.
==> default: [2014-10-19T10:41:21-07:00] INFO: Loading cookbooks [learn_chef_iis@0.1.0]
==> default: [2014-10-19T10:41:23-07:00] INFO: Storing updated cookbooks/learn_chef_iis/.kitchen.yml in the cache.
==> default:
==> default: [2014-10-19T10:41:23-07:00] INFO: Storing updated cookbooks/learn_chef_iis/Berksfile in the cache.
==> default: [2014-10-19T10:41:23-07:00] INFO: Storing updated cookbooks/learn_chef_iis/metadata.rb in the cache.
==> default: [2014-10-19T10:41:23-07:00] INFO: Storing updated cookbooks/learn_chef_iis/README.md in the cache.
==> default: [2014-10-19T10:41:23-07:00] INFO: Storing updated cookbooks/learn_chef_iis/chefignore in the cache.
==> default: [2014-10-19T10:41:26-07:00] INFO: Storing updated cookbooks/learn_chef_iis/metadata.json in the cache.
==> default:
==> default: [2014-10-19T10:41:26-07:00] INFO: Storing updated cookbooks/learn_chef_iis/recipes/default.rb in the cache.
==> default:
==> default: [2014-10-19T10:44:11-07:00] INFO: powershell_script[Install IIS] ran successfully
==> default: [2014-10-19T10:44:11-07:00] INFO: template[c:\inetpub\wwwroot\Default.htm] created file c:\inetpub\wwwroot\
Default.htm
==> default: [2014-10-19T10:44:11-07:00] INFO: template[c:\inetpub\wwwroot\Default.htm] updated file contents c:\inetpub
\wwwroot\Default.htm
==> default: [2014-10-19T10:44:13-07:00] INFO: Chef Run complete in 184.229983 seconds
==> default:
==> default: [2014-10-19T10:44:13-07:00] INFO: Running report handlers
==> default: [2014-10-19T10:44:13-07:00] INFO: Report handlers complete
==> default: [2014-10-19T10:44:13-07:00] INFO: Sending resource update report (run-id: bab4e283-e188-43b2-98d6-1cf8c737b
b6c)
PS C:\Users\Sam\Documents\vagrant\vagrant-chef-windows>
```

Vagrant has now kicked off a chef-client run with the `learnchefiis` cookbook as its runlist. Once it's finished (and in combination with the [forwarded port](#) we setup earlier) you should now be able to open your favourite browser on your host machine and go to <http://localhost:8080> and see...

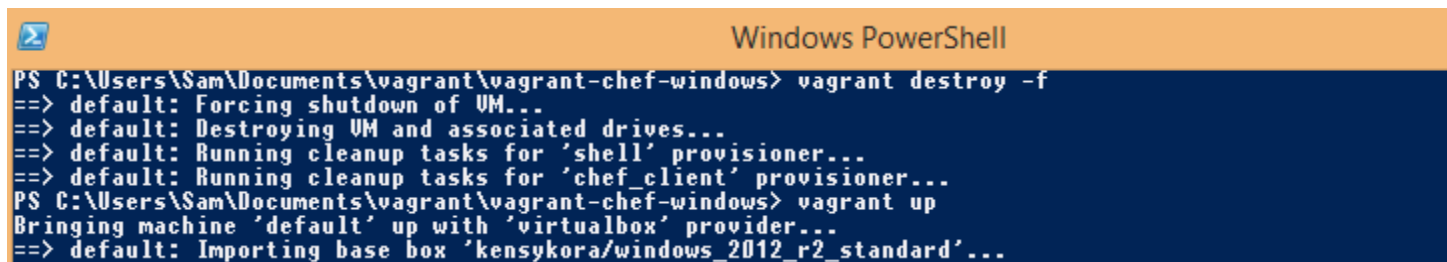


7) Redeploy from Scratch

Now for the moment of truth. Delete the node from the managed Chef environment, destroy the VM and redeploy a fresh one based on the configuration we've provided!



```
vagrant destroy -f vagrant up
```



Wait a little while for Vagrant and Chef to finish doing their thing and you should be able to go back to localhost:8080 again and see exactly the same thing on a fresh VM!