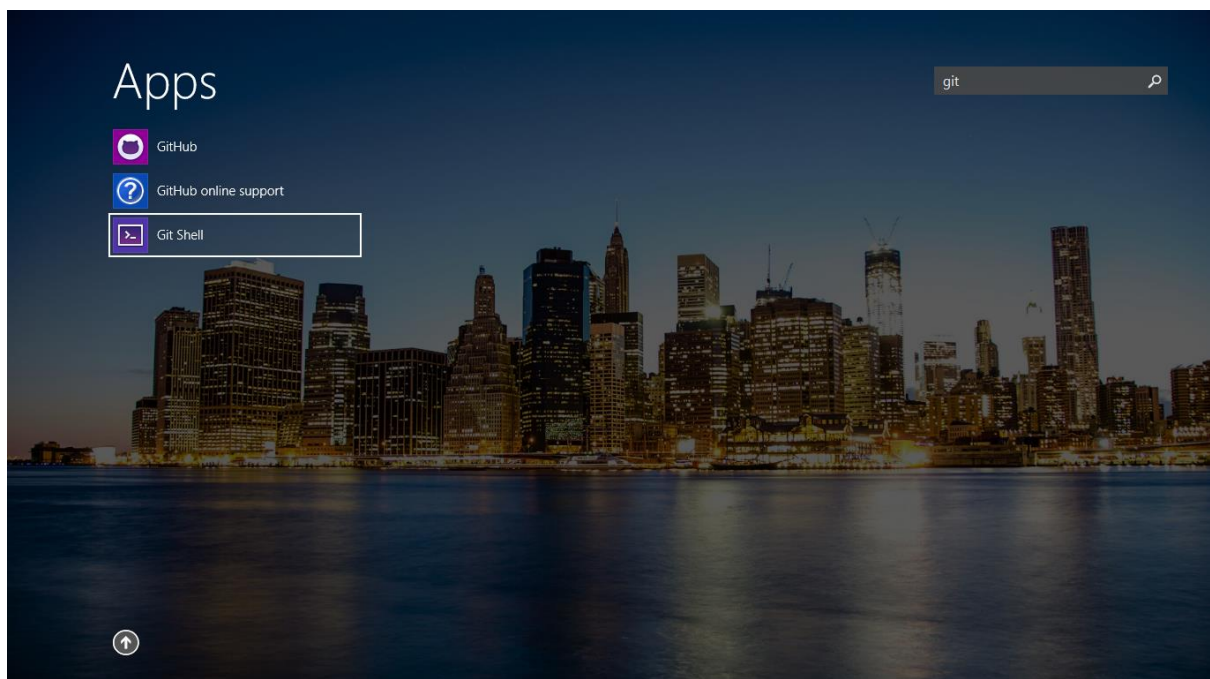


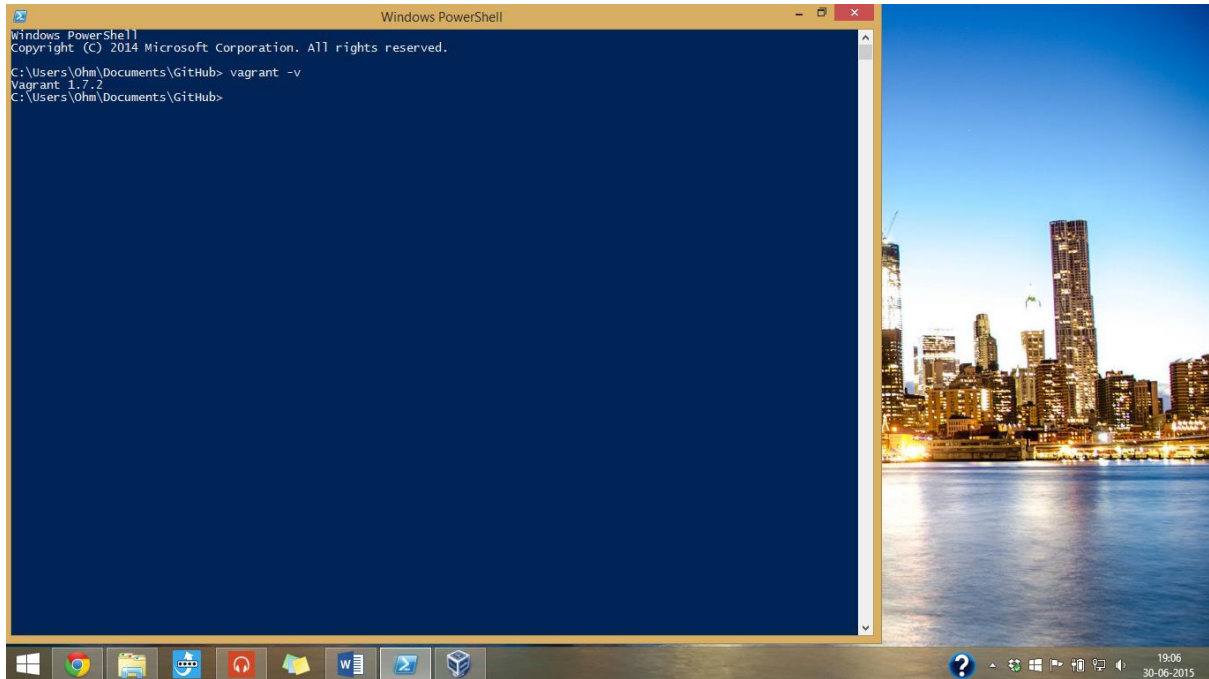
# SMART on FHIR Sandbox Setup

1. Go to <http://docs.smarthealthit.org/sandbox/install/> and from there move forward to the GitHub repository.
2. Refer the file README.md, for all instructions.
3. PRE-REQUISITES: As mentioned in the README file, you will need to install the following
  - a. [Virtual Box](#) – A tool to host virtual machines with your desired guest OS. Version 4.3.20 & up verified and works. Download as per your OS/platform.
  - b. [Vagrant](#) - Vagrant provides easy to configure, reproducible, and portable work environments built on top of industry-standard technology and controlled by a single consistent workflow to help maximize the productivity and flexibility of you and your team. Version 1.7.1 & up verified and works. Download as per your OS/platform.
  - c. [Ansible](#) – Windows users do not need to install this, since it is not supported.

## Installation steps with screenshots (For Windows users)

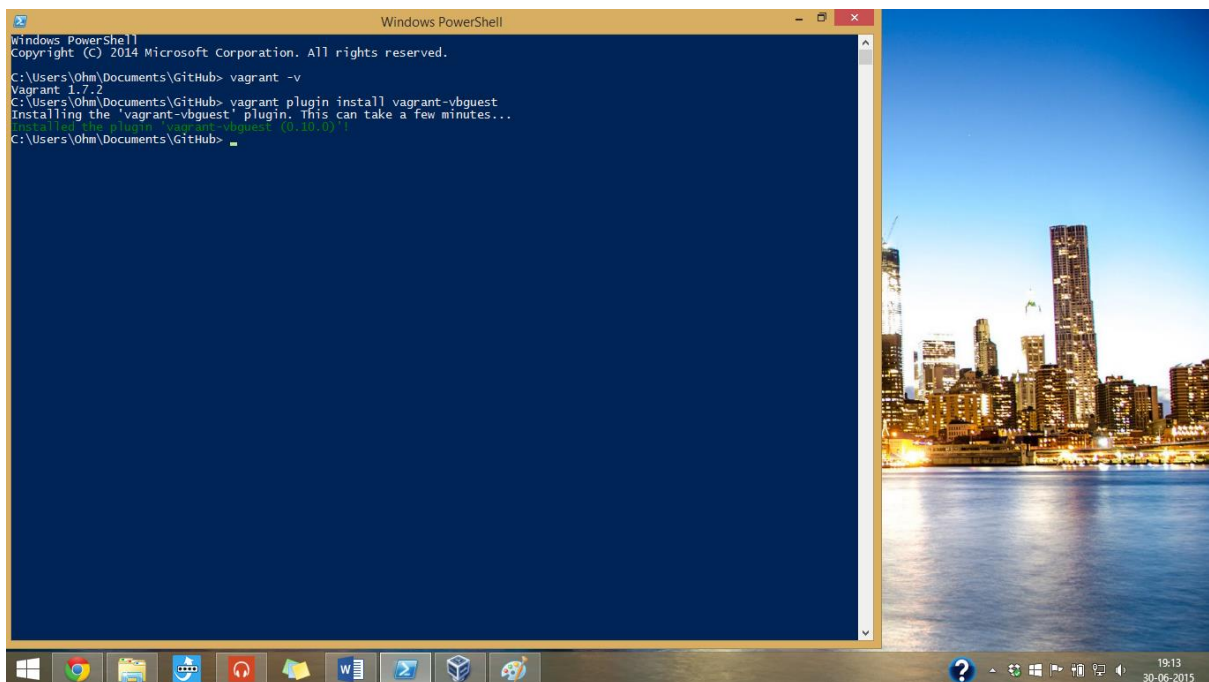
1. Install Virtual Box and launch it.
2. Open GitShell and execute the following commands in it.





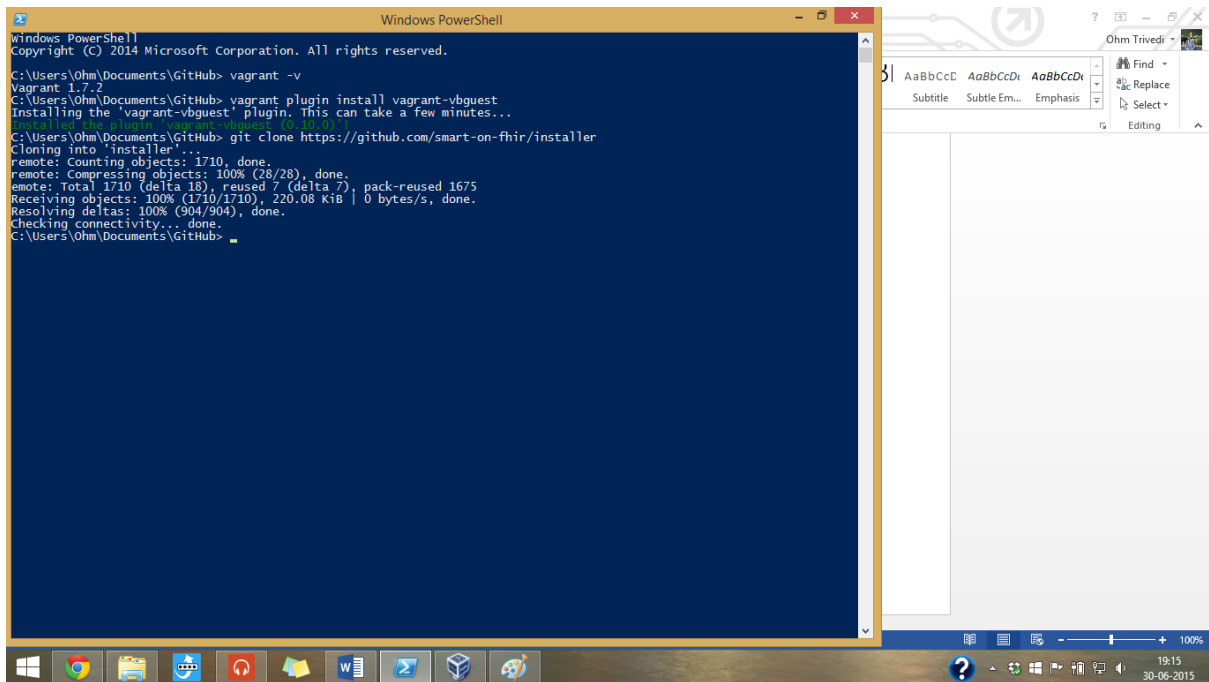
### 3. Install Vagrant plugin **vagrant-vbguest**:

```
vagrant plugin install vagrant-vbguest
```



### 4. Clone the **installer** github repository ([link](https://github.com/smart-on-fhir/installer)):

```
git clone https://github.com/smart-on-fhir/installer
```

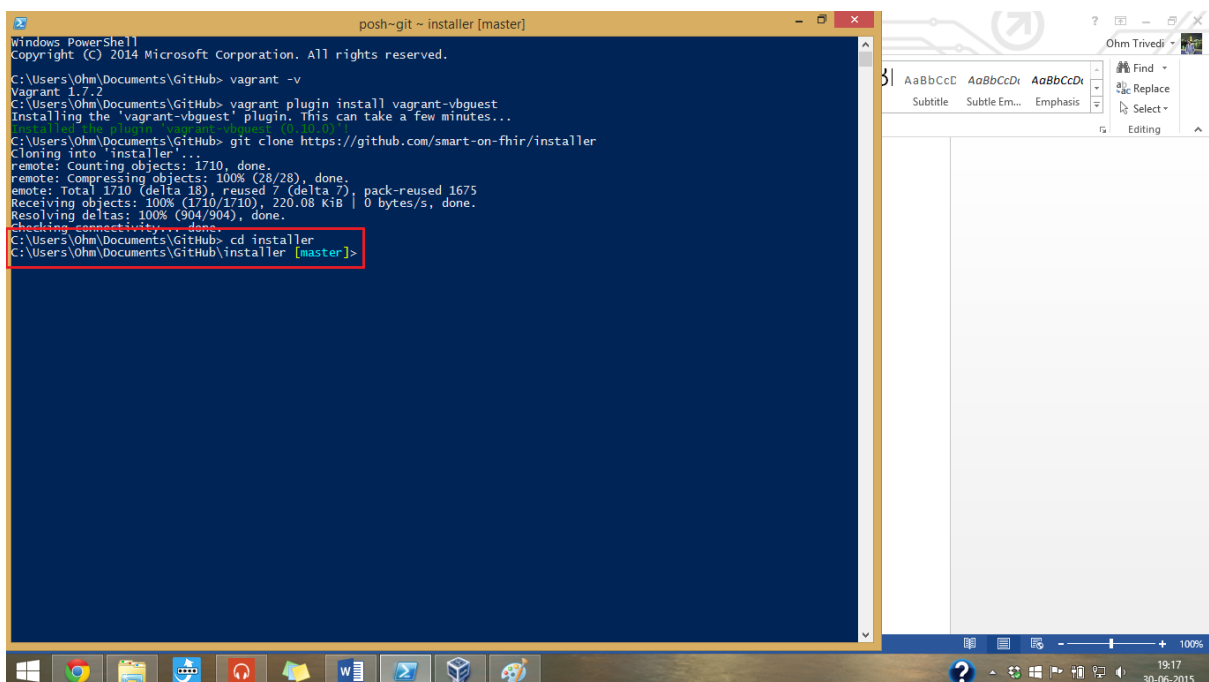


```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

C:\Users\Ohm\Documents\GitHub> vagrant -v
Vagrant 1.7.2
C:\Users\Ohm\Documents\GitHub> vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
C:\Users\Ohm\Documents\GitHub> git clone https://github.com/smart-on-fhir/installer
Cloning into 'installer'...
remote: Counting objects: 1710, done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 1710 (delta 18), reused 7 (delta 7), pack-reused 1675
Receiving objects: 100% (1710/1710), 220.08 KiB | 0 bytes/s, done.
Resolving deltas: 100% (904/904), done.
Checking connectivity... done.
C:\Users\Ohm\Documents\GitHub>
```

5. Change current directory to where **installer** is cloned:

```
cd installer
```

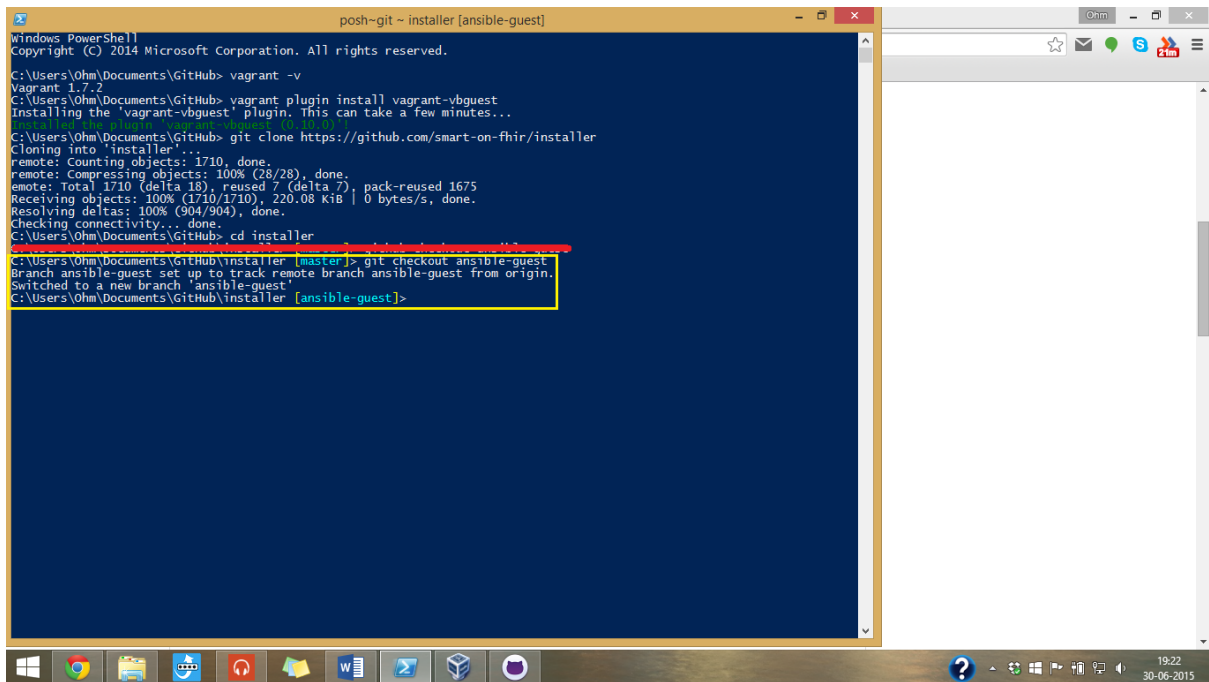


```
posh-git ~ installer [master]
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

C:\Users\Ohm\Documents\GitHub> vagrant -v
Vagrant 1.7.2
C:\Users\Ohm\Documents\GitHub> vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
C:\Users\Ohm\Documents\GitHub> git clone https://github.com/smart-on-fhir/installer
Cloning into 'installer'...
remote: Counting objects: 1710, done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 1710 (delta 18), reused 7 (delta 7), pack-reused 1675
Receiving objects: 100% (1710/1710), 220.08 KiB | 0 bytes/s, done.
Resolving deltas: 100% (904/904), done.
Checking connectivity... done.
C:\Users\Ohm\Documents\GitHub> cd installer
C:\Users\Ohm\Documents\GitHub\installer [master]>
```

6. Change repository's branch from **master** to **ansible-guest**:

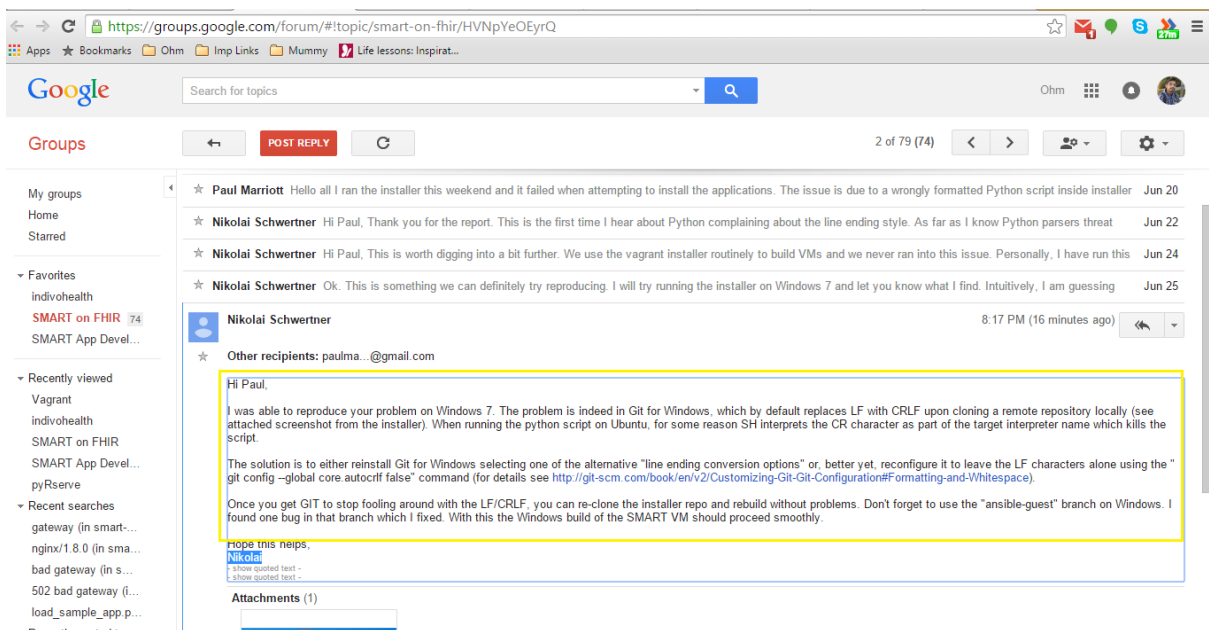
```
git checkout ansible-guest
```



```
posh~git - installer [ansible-guest]
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

C:\Users\Ohm\Documents\GitHub> vagrant -v
Vagrant 1.7.2
C:\Users\Ohm\Documents\GitHub> vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
C:\Users\Ohm\Documents\GitHub> git clone https://github.com/smart-on-fhir/installer
Cloning into 'installer'...
remote: Counting objects: 1710, done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 1710 (delta 18), reused 7 (delta 7), pack-reused 1675
Receiving objects: 100% (1710/1710), 220.08 KiB | 0 bytes/s, done.
Resolving deltas: 100% (904/904), done.
Checking connectivity... done.
C:\Users\Ohm\Documents\GitHub> cd installer
C:\Users\Ohm\Documents\GitHub\installer [master]> git checkout ansible-guest
Branch ansible-guest set up to track remote branch ansible-guest from origin.
Switched to a new branch 'ansible-guest'
C:\Users\Ohm\Documents\GitHub\installer [ansible-guest]>
```

7. There is a bug here (it's related to an automatic action by GitHub Windows client) for which we have a quick fix (steps 8, 9 and 10). You can either go that way **OR** reinstall Git Windows's client using its installer as instructed by Nikolai [here](#). Refer to his 4<sup>th</sup> response. Skip steps 8, 9 and 10 if you choose to go the latter way.



Groups

2 of 79 (74)

★ Paul Marriott Hello all I ran the installer this weekend and it failed when attempting to install the applications. The issue is due to a wrongly formatted Python script inside installer Jun 20

★ Nikolai Schwertner Hi Paul, Thank you for the report. This is the first time I hear about Python complaining about the line ending style. As far as I know Python parsers threat Jun 22

★ Nikolai Schwertner Hi Paul, This is worth digging into a bit further. We use the vagrant installer routinely to build VMs and we never ran into this issue. Personally, I have run this Jun 24

★ Nikolai Schwertner Ok. This is something we can definitely try reproducing. I will try running the installer on Windows 7 and let you know what I find. Intuitively, I am guessing Jun 25

Nikolai Schwertner 8:17 PM (16 minutes ago)

★ Other recipients: paulma...@gmail.com

Hi Paul,

I was able to reproduce your problem on Windows 7. The problem is indeed in Git for Windows, which by default replaces LF with CRLF upon cloning a remote repository locally (see attached screenshot from the installer). When running the python script on Ubuntu, for some reason SH interprets the CR character as part of the target interpreter name which kills the script.

The solution is to either reinstall Git for Windows selecting one of the alternative "line ending conversion options" or, better yet, reconfigure it to leave the LF characters alone using the "git config --global core.autocrlf false" command (for details see <http://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration#Formatting-and-Whitespaces>).

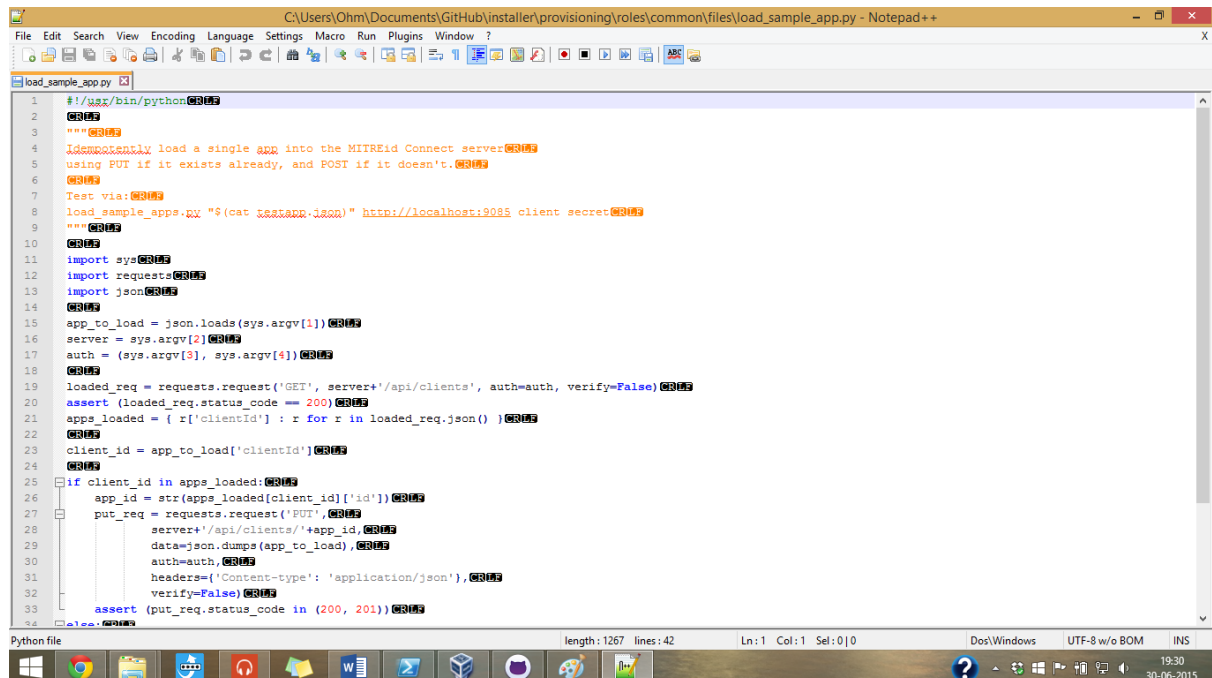
Once you get GIT to stop fooling around with the LF/CRLF, you can re-clone the installer repo and rebuild without problems. Don't forget to use the "ansible-guest" branch on Windows. I found one bug in that branch which I fixed. With this the Windows build of the SMART VM should proceed smoothly.

Hope this helps,

Nikolai

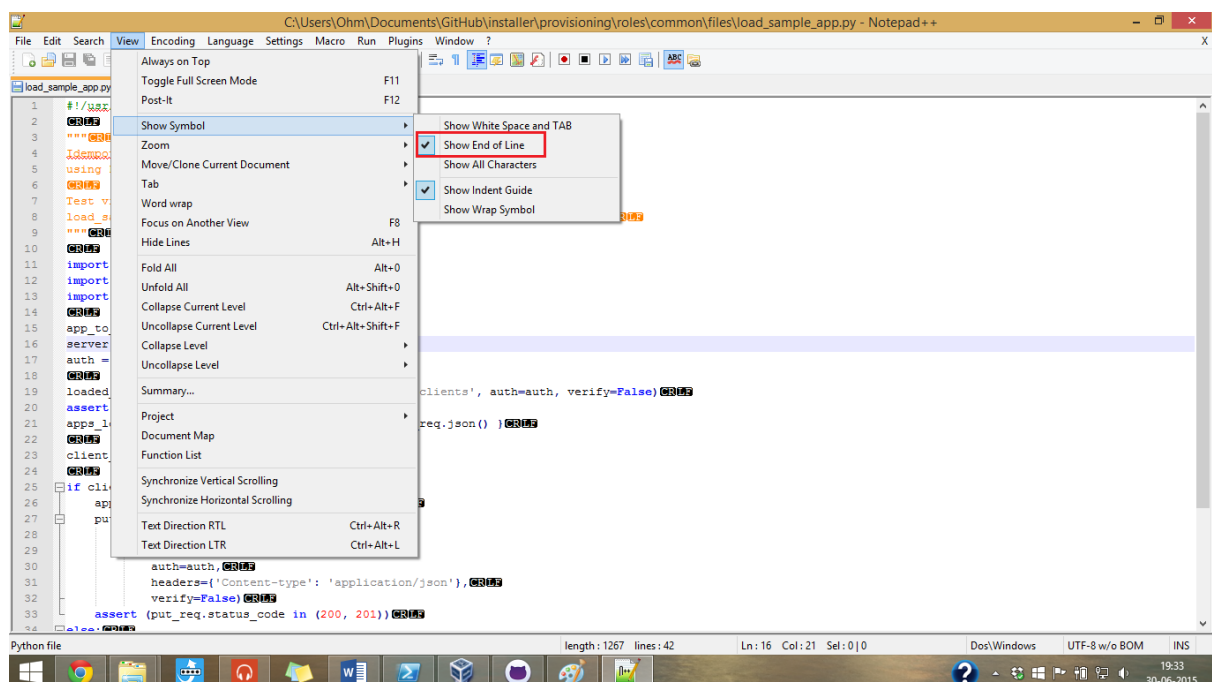
Attachments (1)

8. Open the **installer** github repository in file explorer and open the file **load\_sample\_app.py**. Inside the **installer** directory, it is located here -  
~\GitHub\installer\provisioning\roles\common\files\load\_sample\_app.py  
Open it with an editor like Notepad++ since we need to edit it.



```
1 #!/usr/bin/python
2
3 """
4 Idem:XXXXXX load a single app into the MITREid Connect server
5 using PUT if it exists already, and POST if it doesn't.
6
7 Test via:
8 load_sample_apps.py "$(cat testapp.json)" http://localhost:8085 client secret
9 """
10
11 import sys
12 import requests
13 import json
14
15 app_to_load = json.loads(sys.argv[1])
16 server = sys.argv[2]
17 auth = (sys.argv[3], sys.argv[4])
18
19 loaded_req = requests.request('GET', server+'/api/clients', auth=auth, verify=False)
20 assert (loaded_req.status_code == 200)
21 apps_loaded = [ r['clientId'] : r for r in loaded_req.json() ]
22
23 client_id = app_to_load['clientId']
24
25 if client_id in apps_loaded:
26     app_id = str(apps_loaded[client_id]['id'])
27     put_req = requests.request('PUT',
28                               server+'/api/clients/'+app_id,
29                               data=json.dumps(app_to_load),
30                               auth=auth,
31                               headers={'Content-type': 'application/json'},
32                               verify=False)
33     assert (put_req.status_code in (200, 201))
34 else:
```

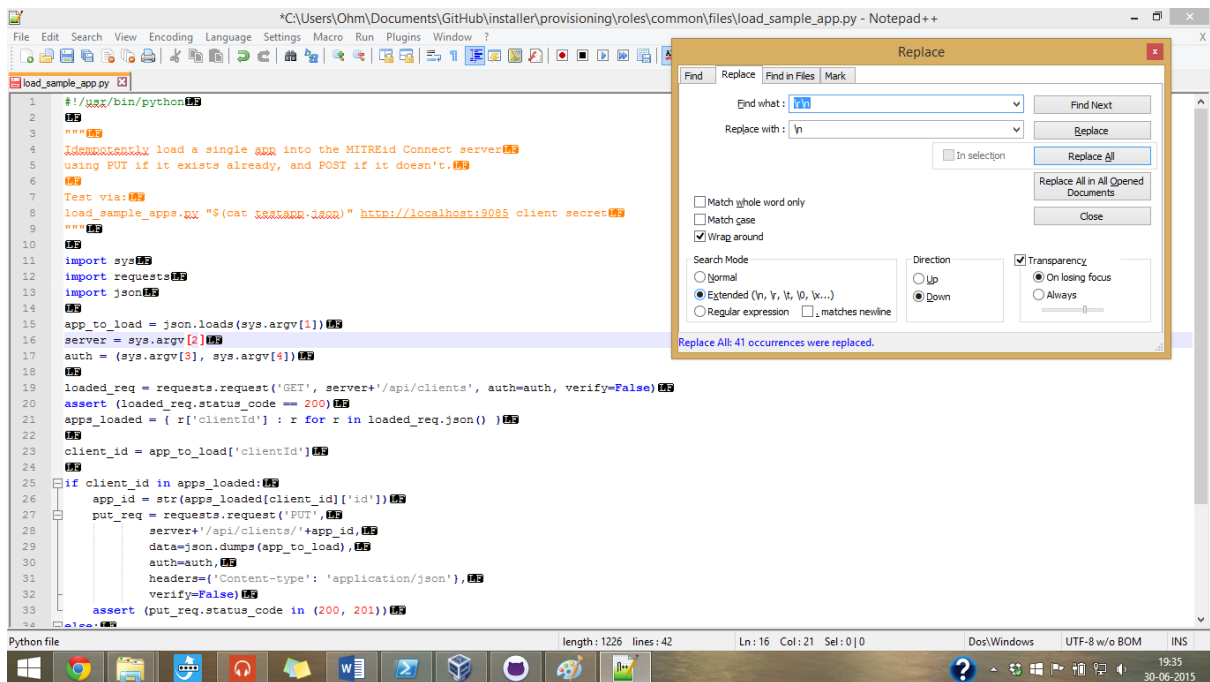
**NOTE:** As seen in screenshot, if you don't see these two characters **CR LF** at the end of every line, then under **View** menu, go to **Show Symbol** and then check **Show End of Line**.



```
1 #!/usr
2
3 """
4 Idem:XXXXXX load a single app into the MITREid Connect server
5 using
6
7 Test v
8 load p
9 """
10
11 import
12 import
13 import
14
15 app_to
16 server
17 auth =
18
19 loaded
20 assert
21 apps_l
22
23 client
24
25 if cli
26     ap
27     pu
28
29
30     auth=auth,
31     headers=('Content-type': 'application/json'),
32     verify=False)
33
34     assert (put_req.status_code in (200, 201))
35 else:
```

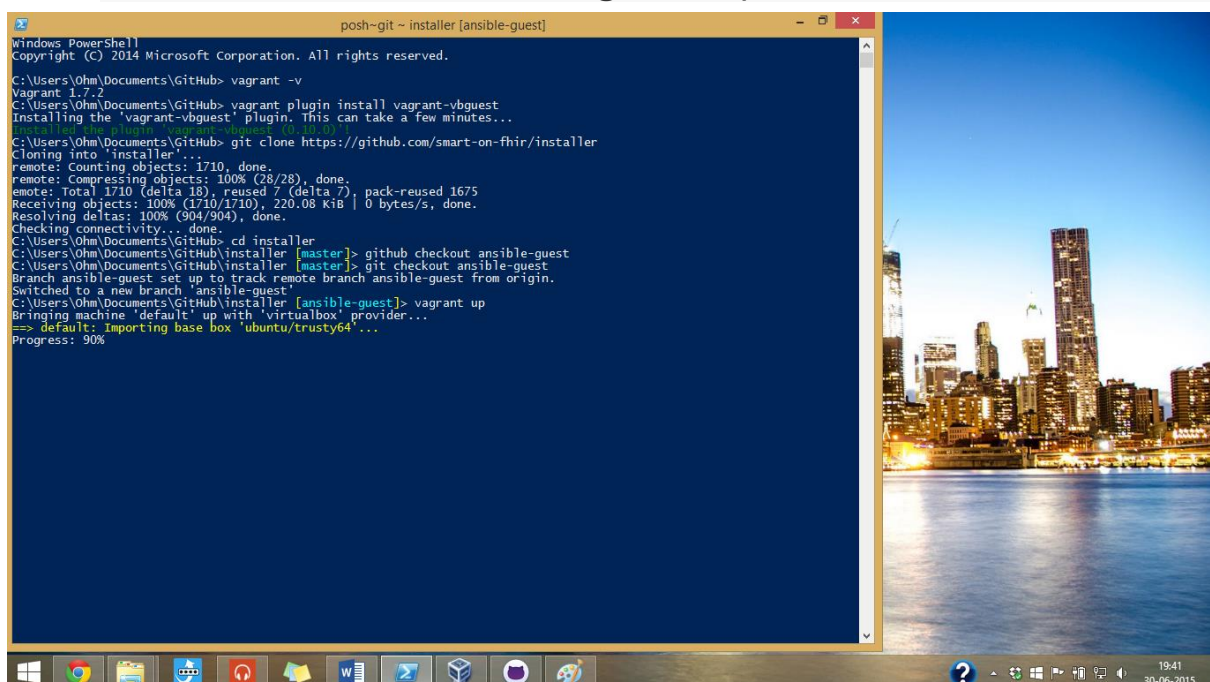


9. Replace all **CR LF** with only **LF**. CR corresponds to \r and LF corresponds to \n. Fill in the following entries in **Replace** dialog box –
- Find what** -> \r\n
  - Replace with** -> \n
  - Ensure in **Search Mode, Extended** mode is selected.

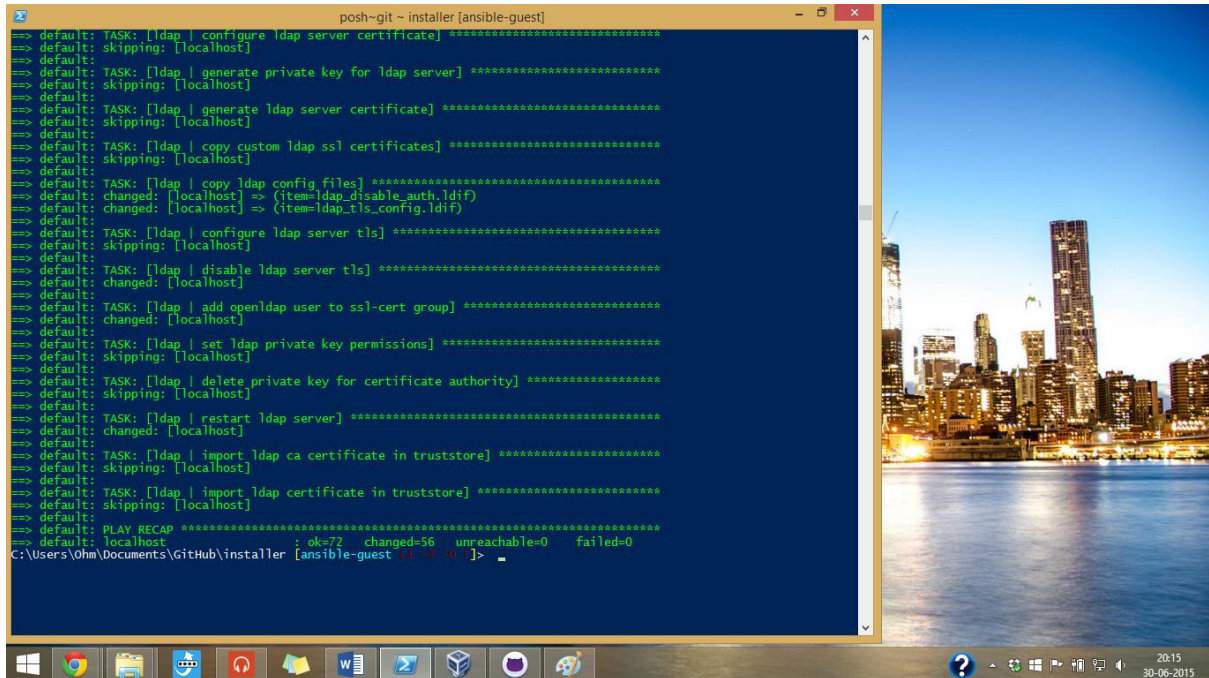


10. Save the file and close.
11. Bring the server online:

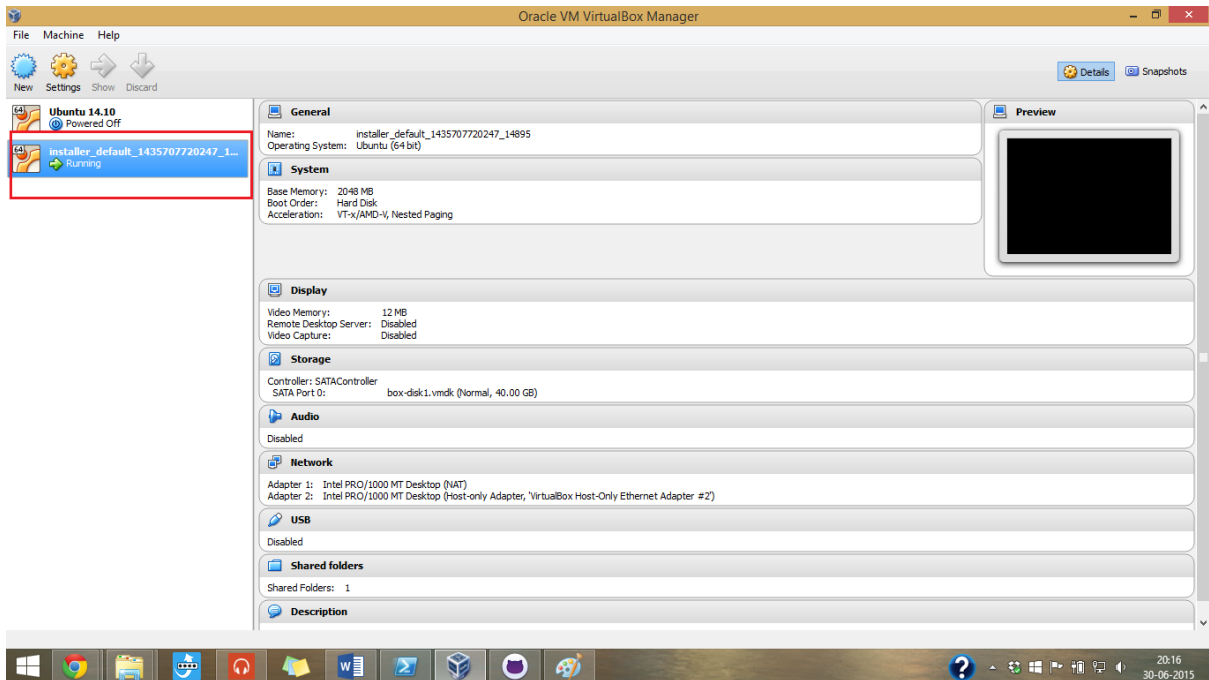
vagrant up



12. The last step takes a few minutes for completion. During the execution, there are some warnings (shown in RED color), you can ignore them. Upon completion, you should see something like this:



13. In Virtual Box, you should have a new VM created and running:

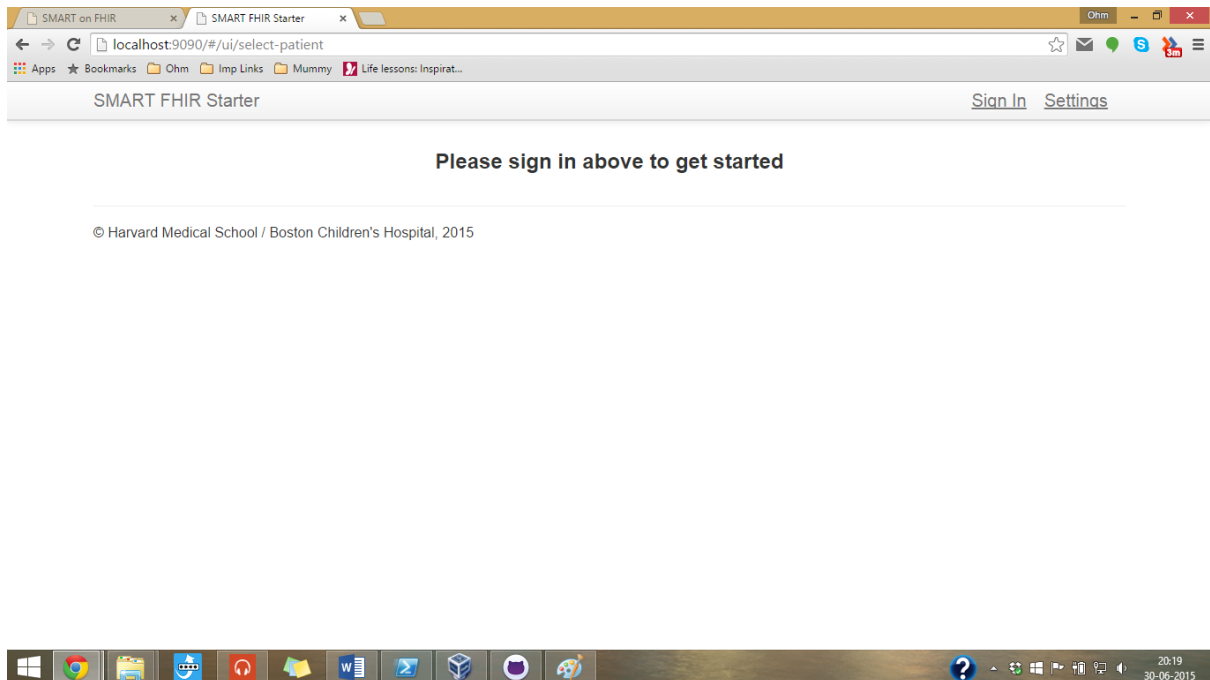


14. Open the following links:

- <http://localhost:9080> for a FHIR API server

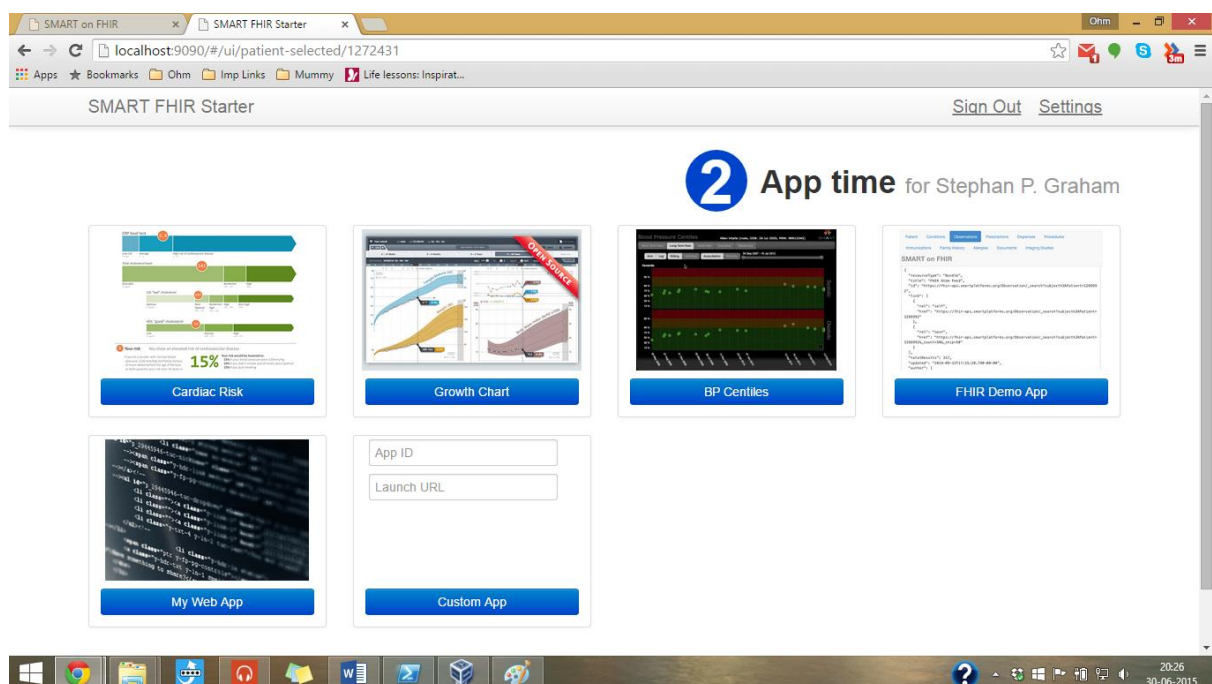
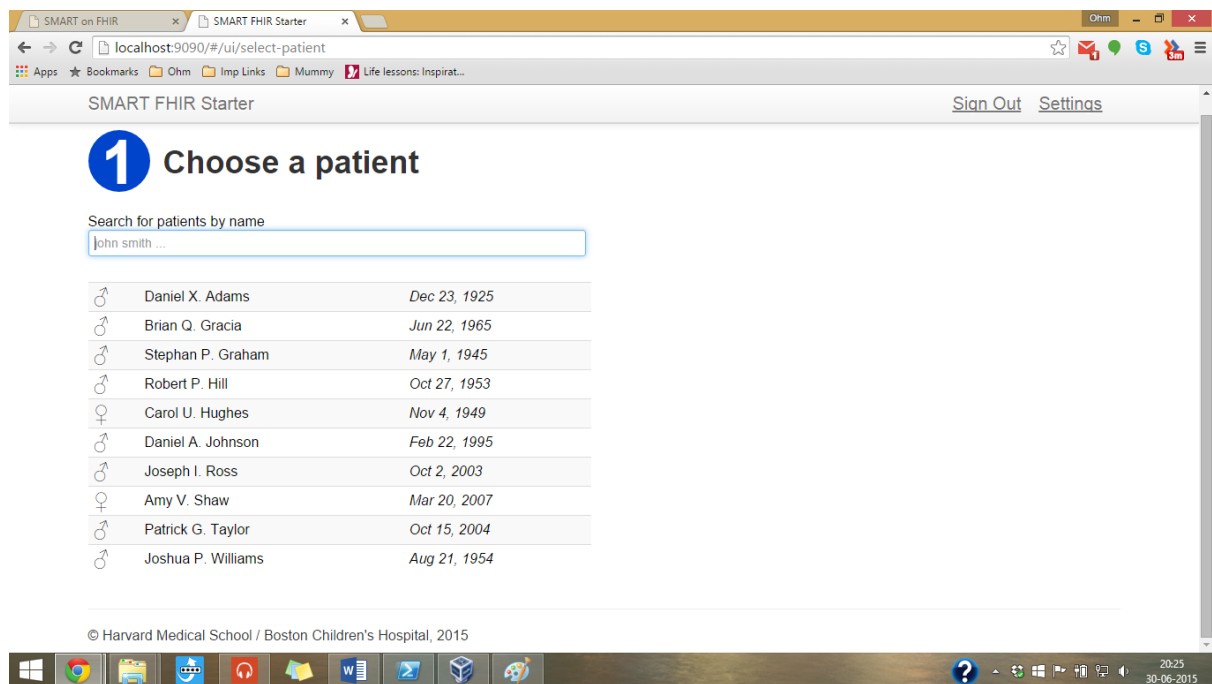


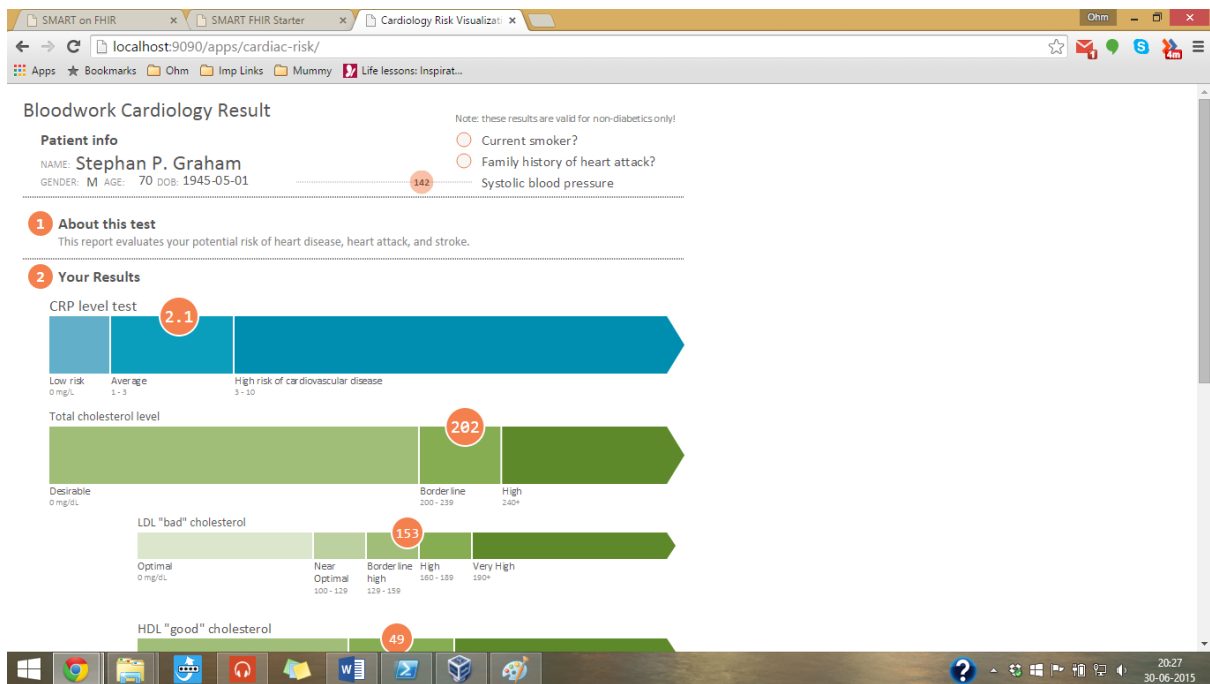
- <http://localhost:9090> for a SMART apps server





15. At **SMART apps server**, you can login using two demo accounts (username/password): **demo/demo** and **admin/password**.
16. After logging in, you should get the option of select any one of the patients, out of 10 patients already loaded as samples. And then you can try out any of the apps for that specific patient. On way you need to give authorizations whenever asked.





17. You can poke around the VM by:

`vagrant ssh`

18. When you are done working, you can shut down the VM with:

`vagrant halt`

```
posh-git ~ installer [ansible-guest]
=> default: TASK: [ldap | configure ldap server certificate] *****
=> default: skipping: [localhost]
=> default: TASK: [ldap | generate private key for ldap server] *****
=> default: skipping: [localhost]
=> default: TASK: [ldap | generate ldap server certificate] *****
=> default: skipping: [localhost]
=> default: TASK: [ldap | copy custom ldap ssl certificates] *****
=> default: skipping: [localhost]
=> default: TASK: [ldap | copy ldap config files] *****
=> default: changed: [localhost] => (item=ldap_disable_auth.ldif)
=> default: changed: [localhost] => (item=ldap_tls_config.ldif)
=> default: TASK: [ldap | configure ldap server tls] *****
=> default: skipping: [localhost]
=> default: TASK: [ldap | disable ldap server tls] *****
=> default: changed: [localhost]
=> default: TASK: [ldap | add openldap user to ssl-cert group] *****
=> default: changed: [localhost]
=> default: TASK: [ldap | set ldap private key permissions] *****
=> default: skipping: [localhost]
=> default: TASK: [ldap | delete private key for certificate authority] *****
=> default: skipping: [localhost]
=> default: TASK: [ldap | restart ldap server] *****
=> default: changed: [localhost]
=> default: TASK: [ldap | import ldap ca certificate in truststore] *****
=> default: skipping: [localhost]
=> default: TASK: [ldap | import ldap certificate in truststore] *****
=> default: skipping: [localhost]
=> default: PLAY RECAP *****
=> default: localhost : ok=72 changed=56 unreachable=0 failed=0
C:\Users\Ohm\Documents\GitHub\installer [ansible-guest (1) (1) (1)]> vagrant halt
=> default: Attempting graceful shutdown of VM...
C:\Users\Ohm\Documents\GitHub\installer [ansible-guest (1) (1) (1)]>
```