Section 07, Mini-Project: Vagrant, Docker, Microservices Example - Windows Cygwin Notes:

Install: bash, curl, python3 (python not needed for this example) - using Cygwin installer

<u>Aliases in .bash_profile:</u>

```
#alias of sudo is needed to run install.sh, other are optional
alias sudo="  #this is 2 double quotes with no spaces between them
alias v='vagrant'
alias c='clear'
alias d='docker'
```

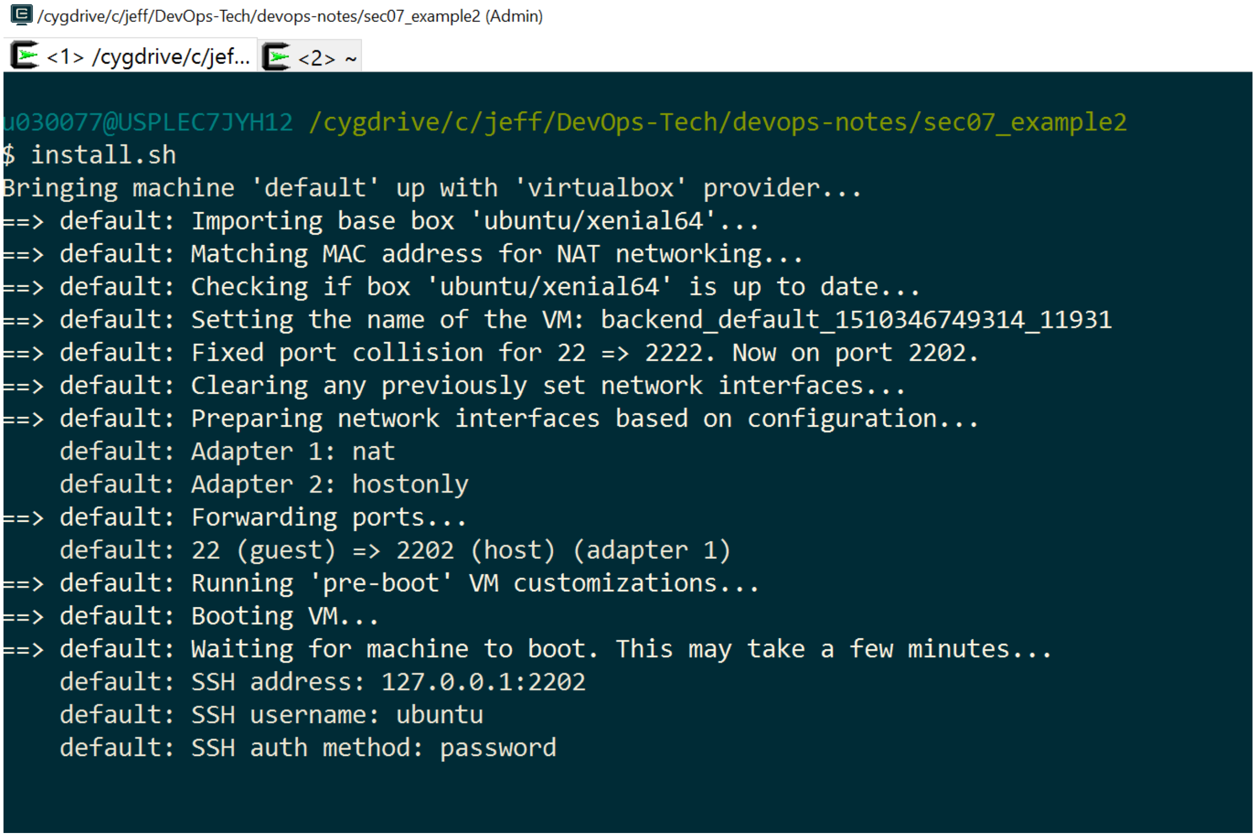
Add "." To the PATH environment variable if it is not there in .bashrc:

```
PATH=.:$PATH
```

After make changes in .bash_profile or .bashrc, to see the changes in the currently open bash shell windows run "source .bash_profile" and or "source .bashrc"

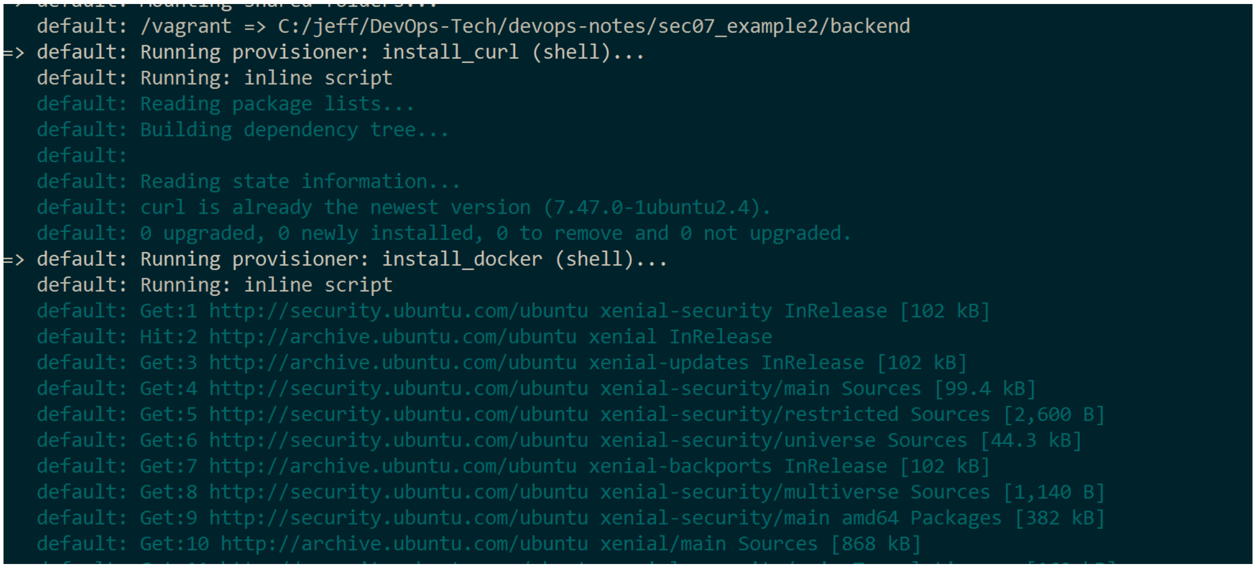
Some screenshots running install.sh in Cygwin:

Backend Install:

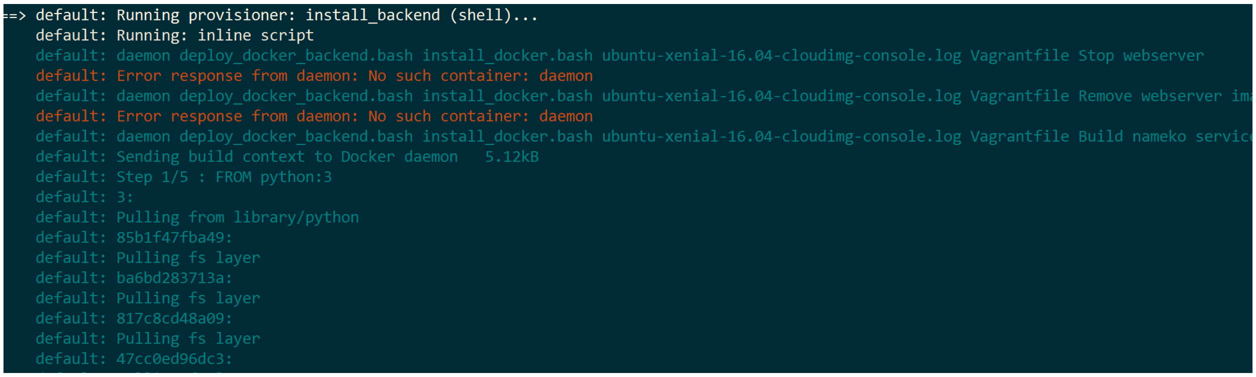


/cygdrive/c/jeff/DevOps-Tech/devops-notes/sec07_example2 (Admin)
<1> /cygdrive/c/jef...
<2> ~

```
u030077@USPLEC7JYH12 /cygdrive/c/jeff/DevOps-Tech/devops-notes/sec07_example2
$ install.sh
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/xenial64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/xenial64' is up to date...
==> default: Setting the name of the VM: backend_default_1510346749314_11931
==> default: Fixed port collision for 22 => 2222. Now on port 2202.
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
    default: Adapter 2: hostonly
==> default: Forwarding ports...
    default: 22 (guest) => 2202 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2202
    default: SSH username: ubuntu
    default: SSH auth method: password
```

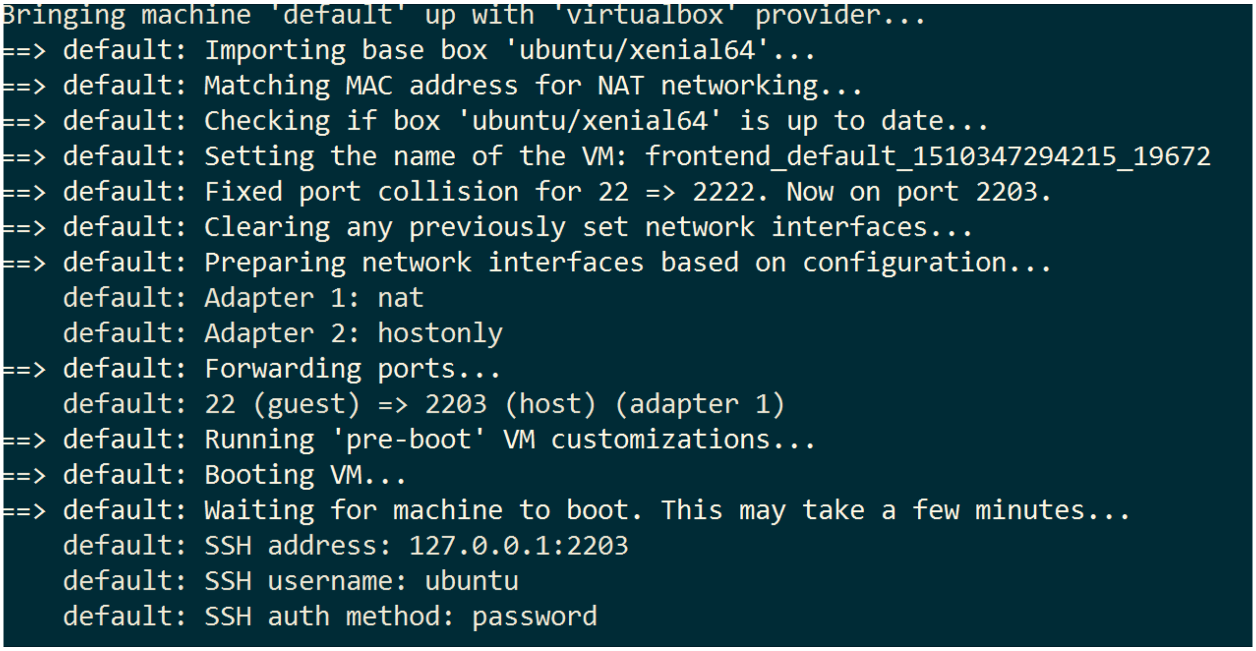


```
=> default: Mounting shared folders...
   default: /vagrant => C:/jeff/DevOps-Tech/devops-notes/sec07_example2/backend
=> default: Running provisioner: install_curl (shell)...
   default: Running: inline script
   default: Reading package lists...
   default: Building dependency tree...
   default:
   default: Reading state information...
   default: curl is already the newest version (7.47.0-1ubuntu2.4).
   default: 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
=> default: Running provisioner: install_docker (shell)...
   default: Running: inline script
   default: Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
   default: Hit:2 http://archive.ubuntu.com/ubuntu xenial InRelease
   default: Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
   default: Get:4 http://security.ubuntu.com/ubuntu xenial-security/main Sources [99.4 kB]
   default: Get:5 http://security.ubuntu.com/ubuntu xenial-security/restricted Sources [2,600 B]
   default: Get:6 http://security.ubuntu.com/ubuntu xenial-security/universe Sources [44.3 kB]
   default: Get:7 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
   default: Get:8 http://security.ubuntu.com/ubuntu xenial-security/multiverse Sources [1,140 B]
   default: Get:9 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [382 kB]
   default: Get:10 http://archive.ubuntu.com/ubuntu xenial/main Sources [868 kB]
```

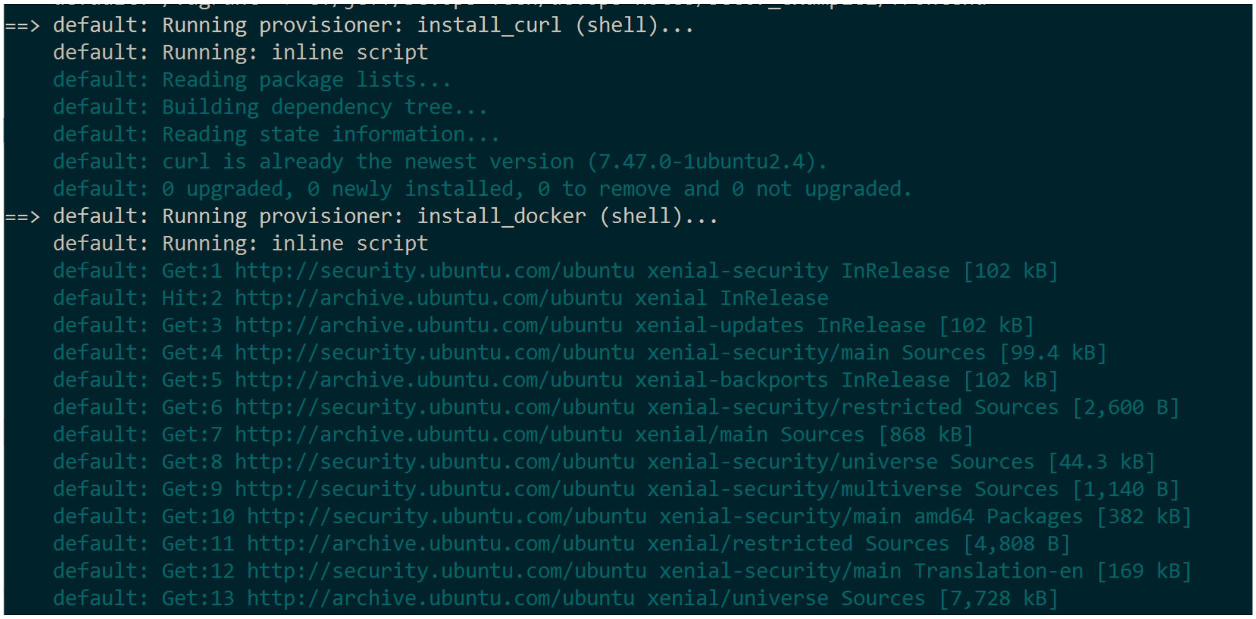


```
==> default: Running provisioner: install_backend (shell)...
    default: Running: inline script
    default: daemon deploy_docker_backend.bash install_docker.bash ubuntu-xenial-16.04-cloudimg-console.log Vagrantfile Stop webserver
    default: Error response from daemon: No such container: daemon
    default: daemon deploy_docker_backend.bash install_docker.bash ubuntu-xenial-16.04-cloudimg-console.log Vagrantfile Remove webserver ima
    default: Error response from daemon: No such container: daemon
    default: daemon deploy_docker_backend.bash install_docker.bash ubuntu-xenial-16.04-cloudimg-console.log Vagrantfile Build nameko service
    default: Sending build context to Docker daemon   5.12kB
    default: Step 1/5 : FROM python:3
    default: 3:
    default: Pulling from library/python
    default: 85b1f47fba49:
    default: Pulling fs layer
    default: ba6bd283713a:
    default: Pulling fs layer
    default: 817c8cd48a09:
    default: Pulling fs layer
    default: 47cc0ed96dc3:
```

FrontEnd Install

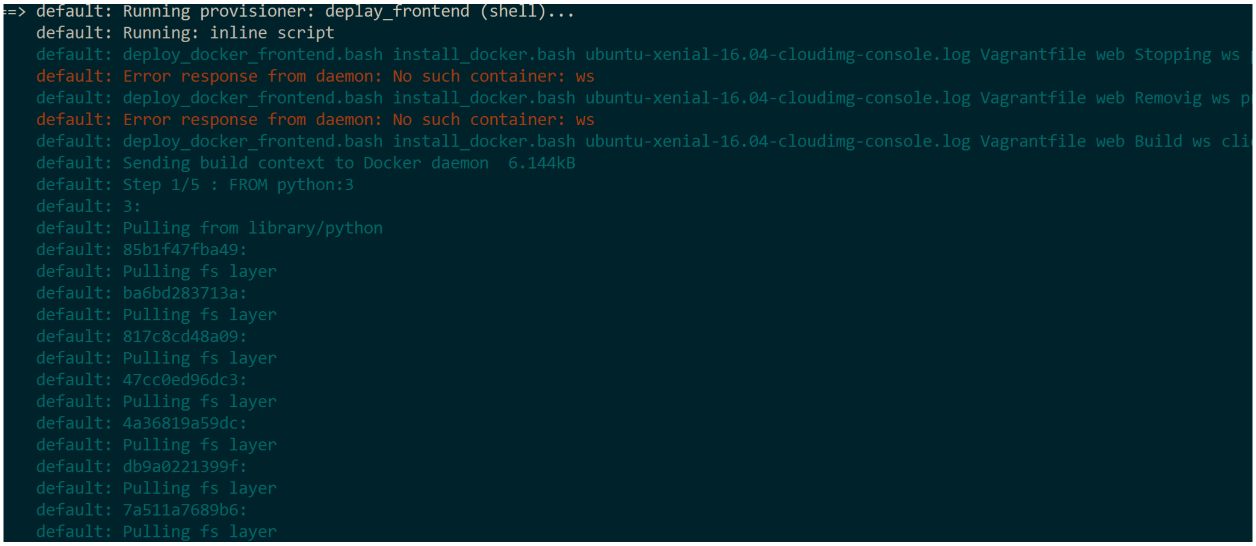


```
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/xenial64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/xenial64' is up to date...
==> default: Setting the name of the VM: frontend_default_1510347294215_19672
==> default: Fixed port collision for 22 => 2222. Now on port 2203.
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
    default: Adapter 2: hostonly
==> default: Forwarding ports...
    default: 22 (guest) => 2203 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2203
    default: SSH username: ubuntu
    default: SSH auth method: password
```



```
==> default: Running provisioner: install_curl (shell)...
    default: Running: inline script
    default: Reading package lists...
    default: Building dependency tree...
    default: Reading state information...
    default: curl is already the newest version (7.47.0-1ubuntu2.4).
    default: 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
==> default: Running provisioner: install_docker (shell)...
    default: Running: inline script
    default: Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
    default: Hit:2 http://archive.ubuntu.com/ubuntu xenial InRelease
    default: Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
    default: Get:4 http://security.ubuntu.com/ubuntu xenial-security/main Sources [99.4 kB]
    default: Get:5 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
    default: Get:6 http://security.ubuntu.com/ubuntu xenial-security/restricted Sources [2,600 B]
    default: Get:7 http://archive.ubuntu.com/ubuntu xenial/main Sources [868 kB]
    default: Get:8 http://security.ubuntu.com/ubuntu xenial-security/universe Sources [44.3 kB]
    default: Get:9 http://security.ubuntu.com/ubuntu xenial-security/multiverse Sources [1,140 B]
    default: Get:10 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [382 kB]
    default: Get:11 http://archive.ubuntu.com/ubuntu xenial/restricted Sources [4,808 B]
    default: Get:12 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [169 kB]
    default: Get:13 http://archive.ubuntu.com/ubuntu xenial/universe Sources [7,728 kB]
```

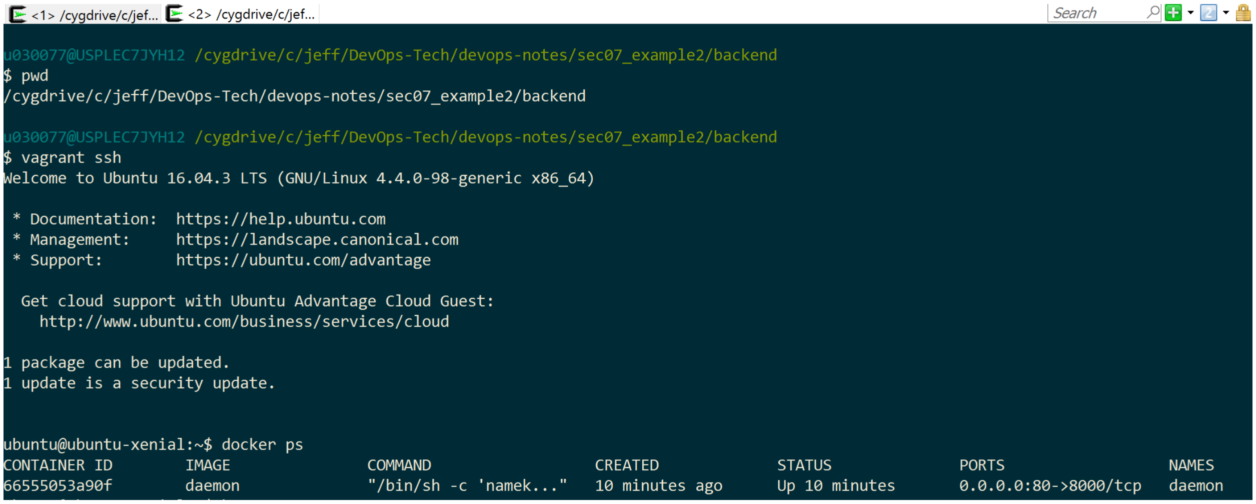
NOTE: I misspelled provisioner name deploy_docker as deplay_docker in the slide below – since the name of an inline bash script vagrant provisioner is just for us – this does not matter



```
==> default: Running provisioner: deplay_frontend (shell)...
    default: Running: inline script
    default: deploy_docker_frontend.bash install_docker.bash ubuntu-xenial-16.04-cloudimg-console.log Vagrantfile web Stopping ws
    default: Error response from daemon: No such container: ws
    default: deploy_docker_frontend.bash install_docker.bash ubuntu-xenial-16.04-cloudimg-console.log Vagrantfile web Removig ws p
    default: Error response from daemon: No such container: ws
    default: deploy_docker_frontend.bash install_docker.bash ubuntu-xenial-16.04-cloudimg-console.log Vagrantfile web Build ws cli
    default: Sending build context to Docker daemon  6.144kB
    default: Step 1/5 : FROM python:3
    default: 3:
    default: Pulling from library/python
    default: 85b1f47fba49:
    default: Pulling fs layer
    default: ba6bd283713a:
    default: Pulling fs layer
    default: 817c8cd48a09:
    default: Pulling fs layer
    default: 47cc0ed96dc3:
    default: Pulling fs layer
    default: 4a36819a59dc:
    default: Pulling fs layer
    default: db9a0221399f:
    default: Pulling fs layer
    default: 7a511a7689b6:
    default: Pulling fs layer
```

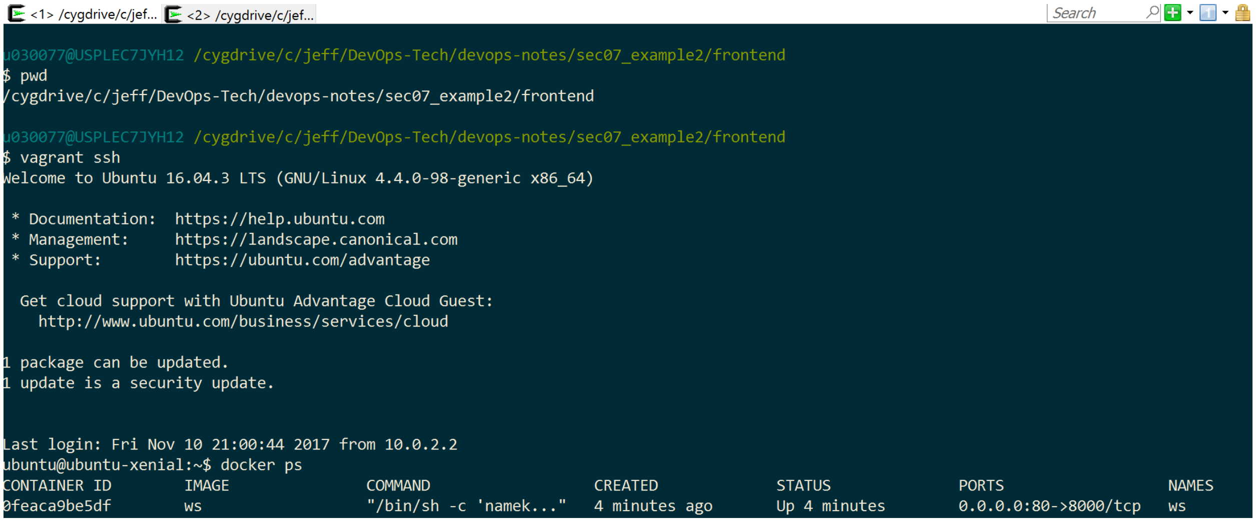
After install.sh finished:

Manually, verify the backend is running – normally we would automate this step:



```
u030077@USPLEC7JYH12 /cygdrive/c/jeff/DevOps-Tech/devops-notes/sec07_example2/backend
$ pwd
/cygdrive/c/jeff/DevOps-Tech/devops-notes/sec07_example2/backend

u030077@USPLEC7JYH12 /cygdrive/c/jeff/DevOps-Tech/devops-notes/sec07_example2/backend
$ vagrant ssh
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-98-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

1 package can be updated.
1 update is a security update.


ubuntu@ubuntu-xenial:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                  NAMES
66555053a90f        daemon              "/bin/sh -c 'namek..."   10 minutes ago      Up 10 minutes       0.0.0.0:80->8000/tcp   daemon
```

Next, manually verify the frontend is running – again we would really want to automate this step:



```
u030077@USPLEC7JYH12 /cygdrive/c/jeff/DevOps-Tech/devops-notes/sec07_example2/frontend
$ pwd
/cygdrive/c/jeff/DevOps-Tech/devops-notes/sec07_example2/frontend

u030077@USPLEC7JYH12 /cygdrive/c/jeff/DevOps-Tech/devops-notes/sec07_example2/frontend
$ vagrant ssh
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-98-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

1 package can be updated.
1 update is a security update.


Last login: Fri Nov 10 21:00:44 2017 from 10.0.2.2
ubuntu@ubuntu-xenial:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                  NAMES
0feaca9be5df        ws                  "/bin/sh -c 'namek..."   4 minutes ago       Up 4 minutes        0.0.0.0:80->8000/tcp   ws
```

If you have problems reconnecting try the following:

a) In one Cygwin Bash console, in backend:

```
vagrant reload #or vagrant halt, vagrant up
vagrant ssh
#in vm
docker ps #you will probably not see anything
docker images #you should see daemon
docker start –i daemon #-i runs the Docker container interactively, you can see its output
```

b) In another Cygwin Bash console in frontend

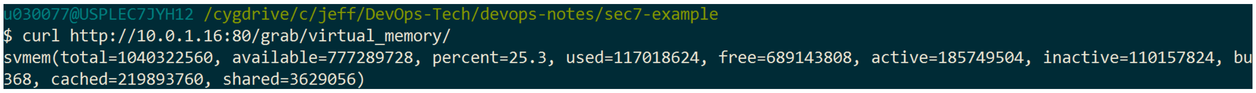
```
vagrant reload #or vagrant halt, vagrant up
vagrant ssh
#in vm
docker ps #you will probably not see anything
docker images #you should see ws
docker start –i ws
```

c) In another Cygwin Bash console

```
curl http://10.0.1.16:80/grab/virtual_memory/
```

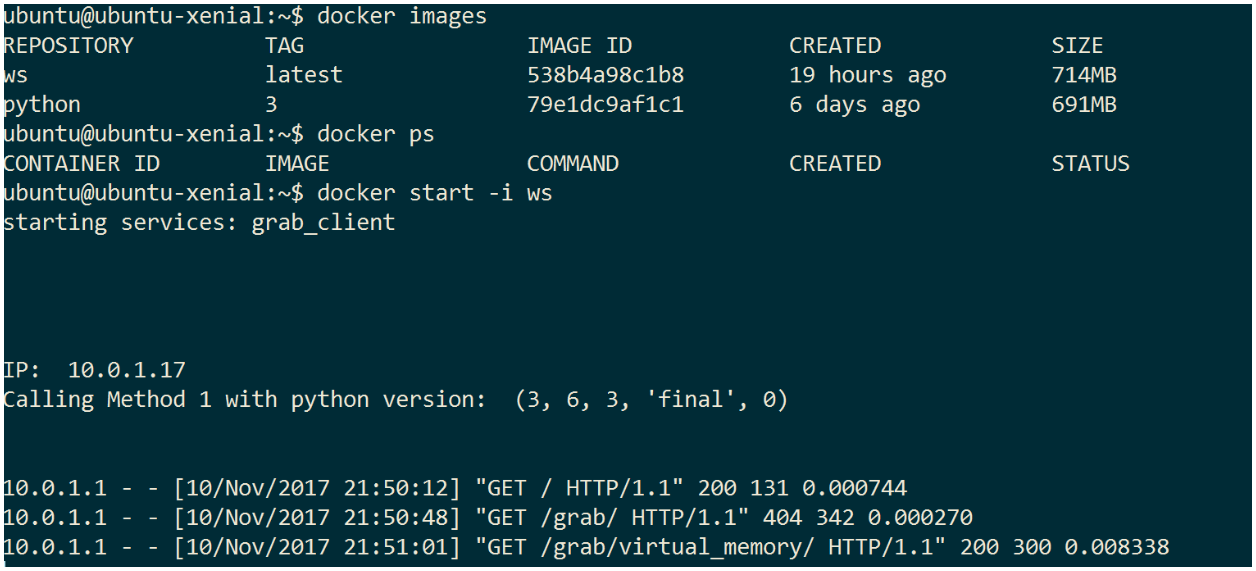
The following images show the client, the frontend, and the backend outputs:

Client, Cygwin Bash Console on Windows 10



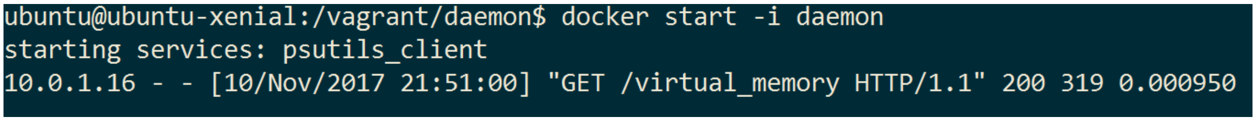
```
u030077@USPLEC7JYH12 /cygdrive/c/jeff/DevOps-Tech/devops-notes/sec7-example
$ curl http://10.0.1.16:80/grab/virtual_memory/
svmem(total=1040322560, available=777289728, percent=25.3, used=117018624, free=689143808, active=185749504, inactive=110157824, bu
368, cached=219893760, shared=3629056)
```

Frontend, ssh from Cygwin Bash on Windows 10



```
ubuntu@ubuntu-xenial:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ws                  latest              538b4a98c1b8        19 hours ago        714MB
python              3                   79e1dc9af1c1        6 days ago          691MB
ubuntu@ubuntu-xenial:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
ubuntu@ubuntu-xenial:~$ docker start -i ws
starting services: grab_client




IP:  10.0.1.17
Calling Method 1 with python version:  (3, 6, 3, 'final', 0)


10.0.1.1 - - [10/Nov/2017 21:50:12] "GET / HTTP/1.1" 200 131 0.000744
10.0.1.1 - - [10/Nov/2017 21:50:48] "GET /grab/ HTTP/1.1" 404 342 0.000270
10.0.1.1 - - [10/Nov/2017 21:51:01] "GET /grab/virtual_memory/ HTTP/1.1" 200 300 0.008338
```

Backend, ssh Cygwin Bash on Windows 10



```
ubuntu@ubuntu-xenial:/vagrant/daemon$ docker start -i daemon
starting services: psutils_client
10.0.1.16 - - [10/Nov/2017 21:51:00] "GET /virtual_memory HTTP/1.1" 200 319 0.000950
```