Instituto de Investigación de Recursos Biológicos
Alexander von Humboldt

MANUAL COMPLETO DE INSTALACIÓN DEL AMBIENTE DE CNCFLORA PARA EL INSTITUTO ALEXANDER VON HUMBOLDT

VERSION 1.0 – 01-12-2016

Elaborado Por: Jaime Alberto Gutiérrez Mejía

ÍNDICE

1. OBJETO DEL DOCUMENTO

Este documento tiene como objeto mostrar una visión del panorama tecnológico de la aplicación CNCFLORA PORTAL y del clúster de aplicaciones NUVEM de la solución, con el fin de conocer cómo descomprimir la paquetería, instalarla y poder ponerla a punto.

2. PARTICIPANTES

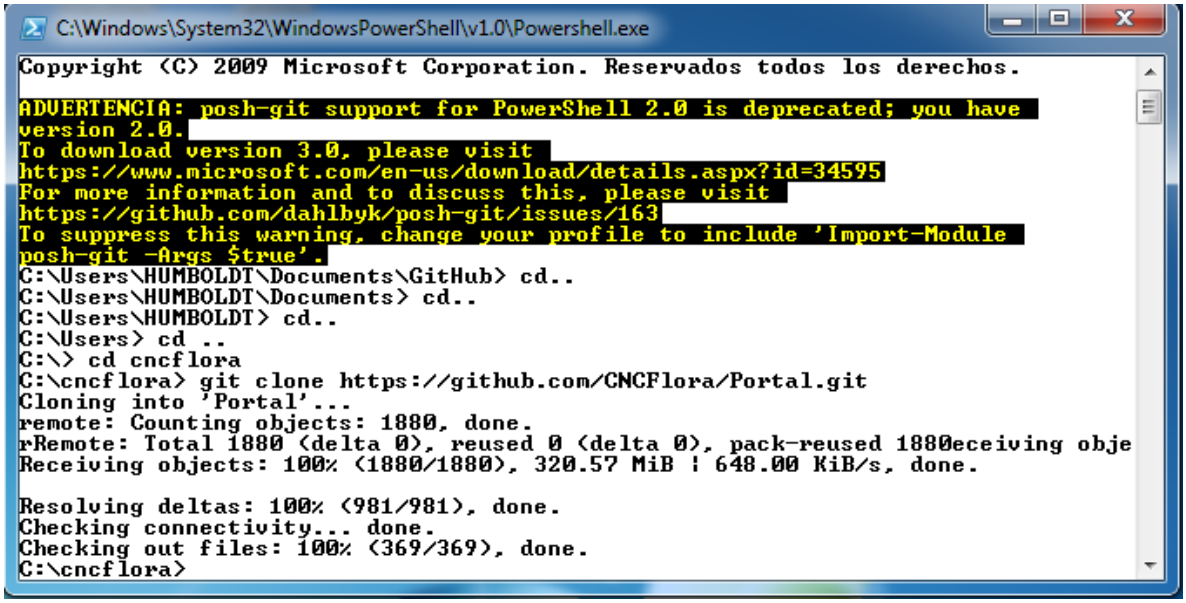
En este proceso tiene principal participación el Administrador Técnico del Sistema, el jefe de gestión de aplicaciones del Instituto Alexander von Humboldt y los participantes del proceso de despliegue y paso a producción.

3. OBJETIVOS

a) El objetivo de este proceso, es asegurar el montaje del ambiente de virtualización
b) Configurar la paquetería base Linux del Sistema Operativo Host de la solución
c) Habilitar las extensiones técnicas para el correcto funcionamiento de todos los módulos
d) Poner la solución a punto y ponerla a funcionar en el ambiente virtualizado escogido

4. INSTALACIÓN DE AMBIENTE VAGRANT, GITSHELL Y LÍNEA BASE DE PYTHON EN LA MÁQUINA VIRTUAL LINUX UBUNTU 14.04 DE LA DISTRIBUCIÓN

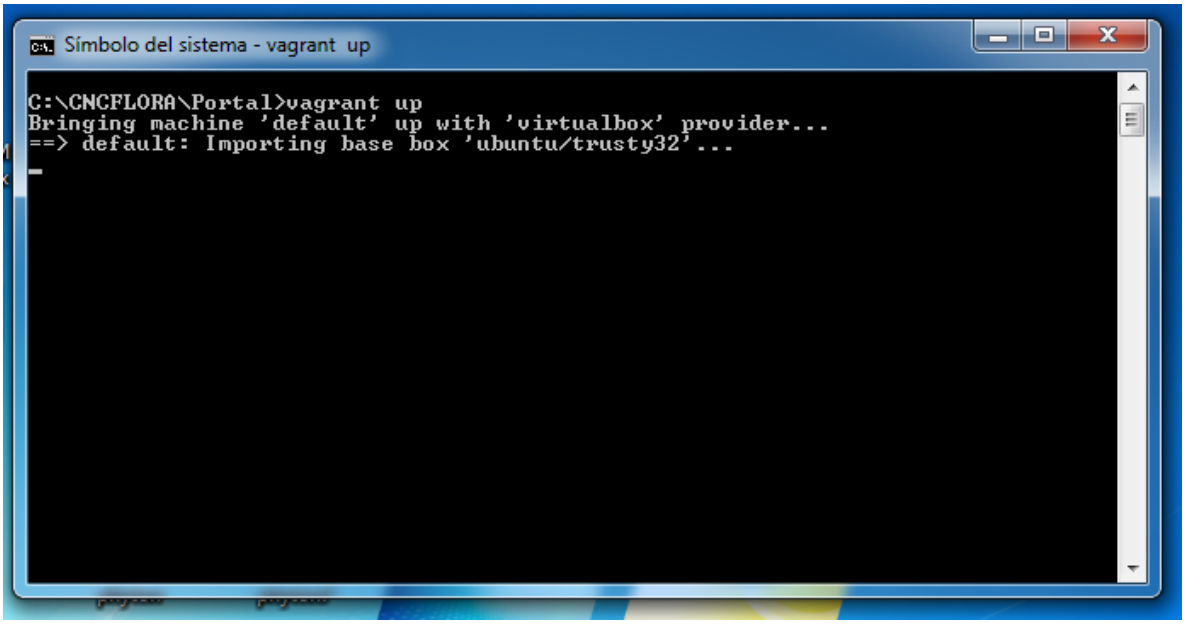
1) Clonar repositorio con el comando git clone https://github.com/CNCFlora/Portal.git

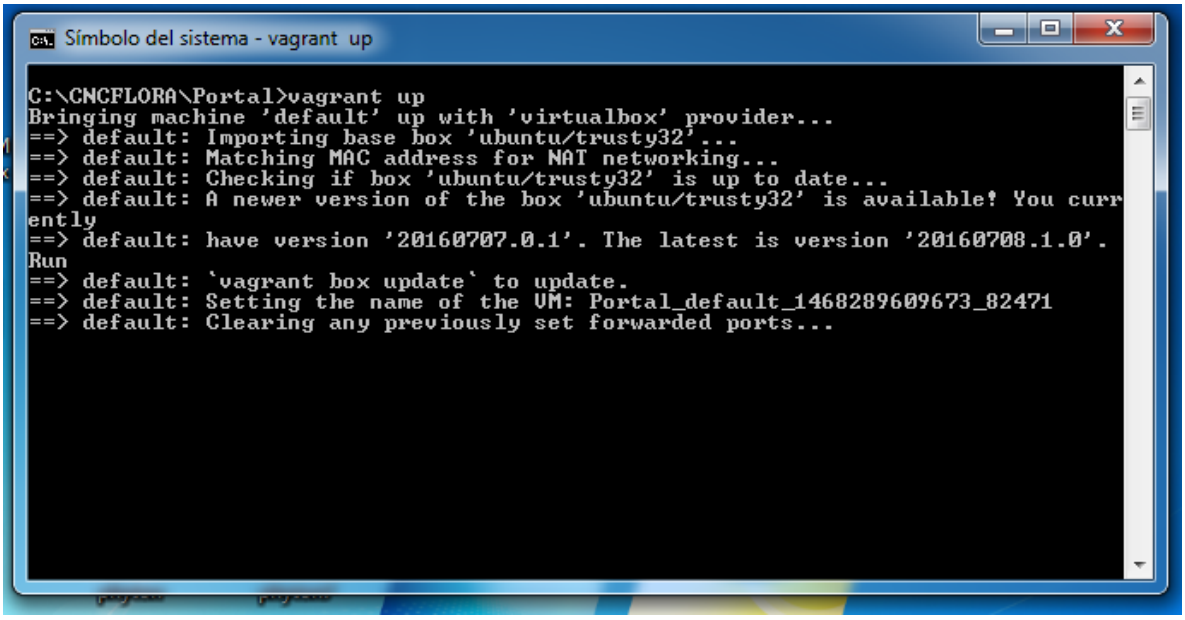


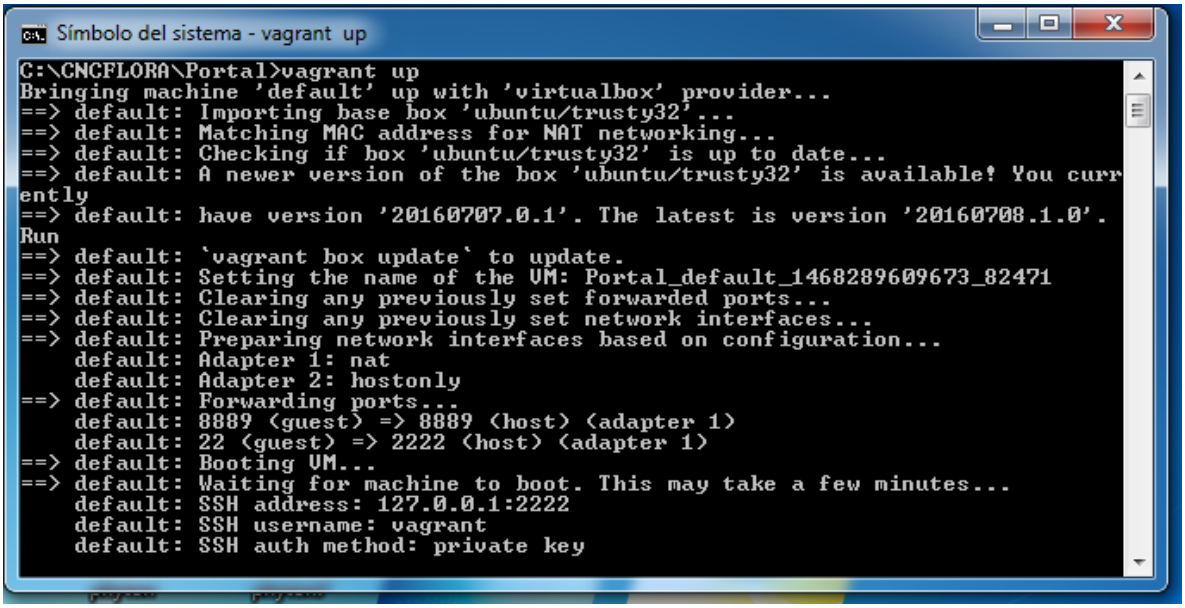
2) Elevar la máquina virtual luego de haber instalado

   a) Virtual Box
   b) Pack de actualizaciones
   c) Microsoft Visual C++ 2010 Redistributable

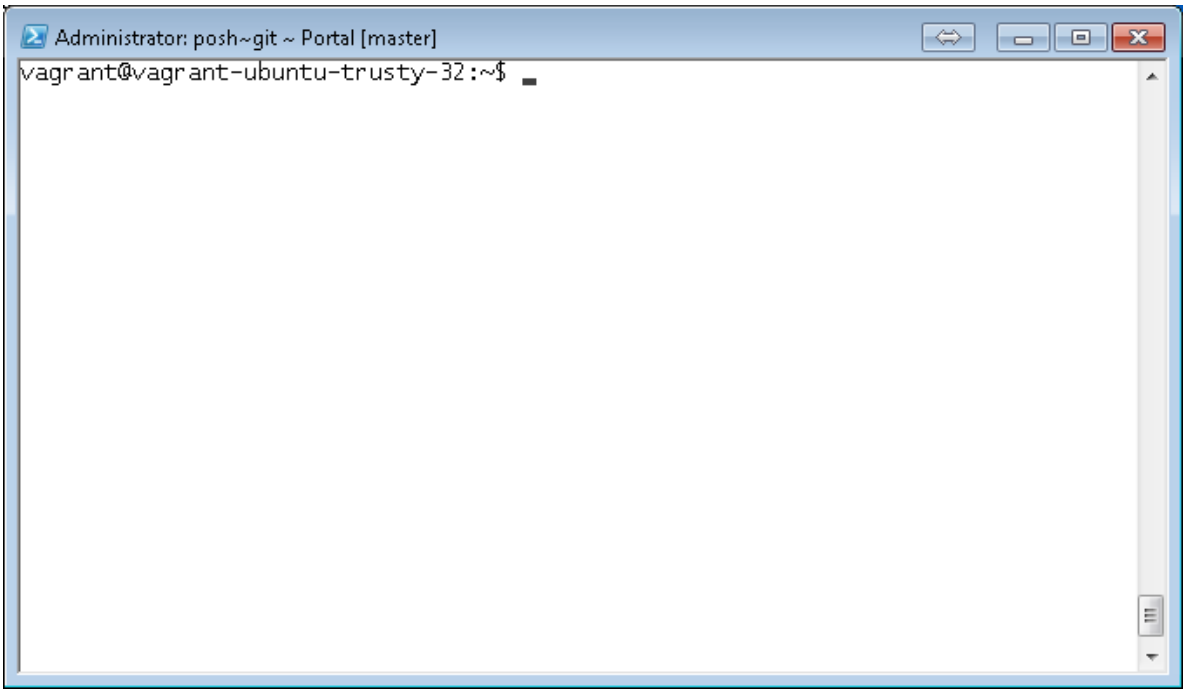
   Comando: vagrant up


Símbolo del sistema - vagrant up
C:\CNCFLORA\Portal>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty32'...

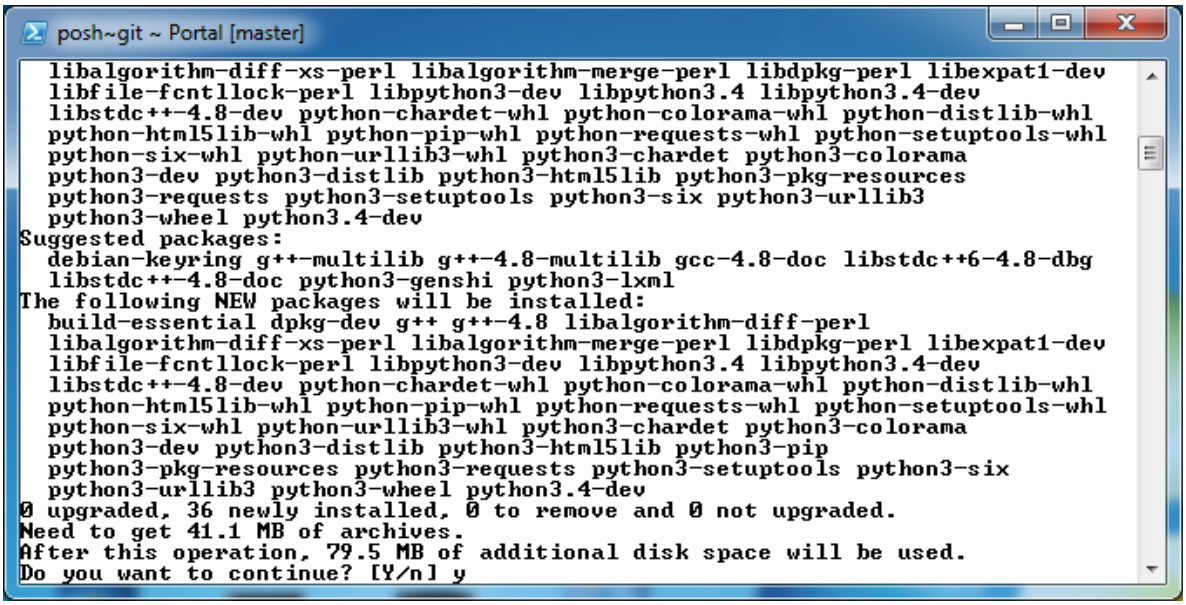

Símbolo del sistema - vagrant up
C:\CNCFLORA\Portal>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty32'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty32' is up to date...
==> default: A newer version of the box 'ubuntu/trusty32' is available! You curr
ently
==> default: have version '20160707.0.1'. The latest is version '20160708.1.0'.
Run
==> default: `vagrant box update` to update.
==> default: Setting the name of the VM: Portal_default_1468289609673_82471
==> default: Clearing any previously set forwarded ports...



3) Una vez la máquina esté arriba, conectarse desde la consola de GitHub con vagrant ssh

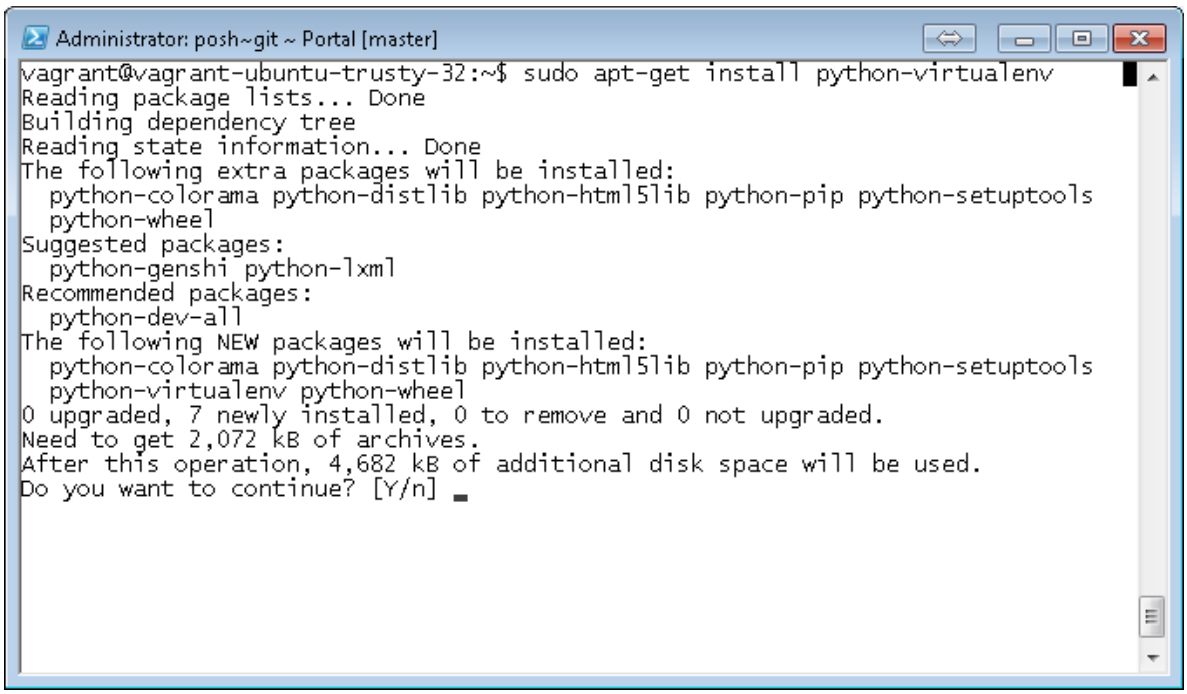


4) Instalar el paquete de PYTHON 3, con el comando
sudo apt-get install python3-pip



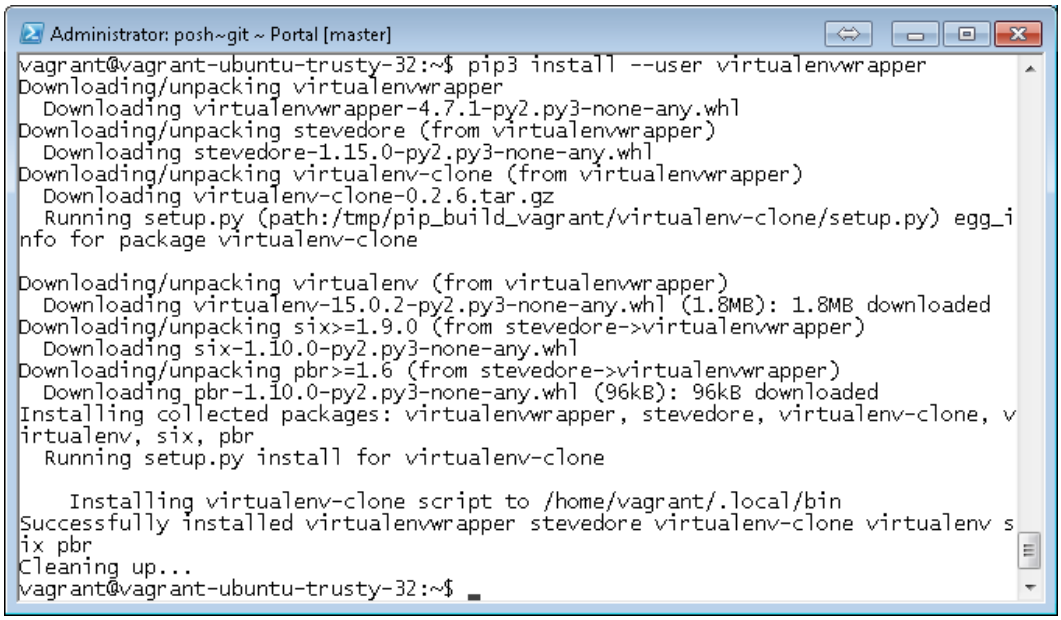
5) Instalar Python virtual env

sudo apt-get install python-virtualenv



6) Bajar el virtualenwrapper

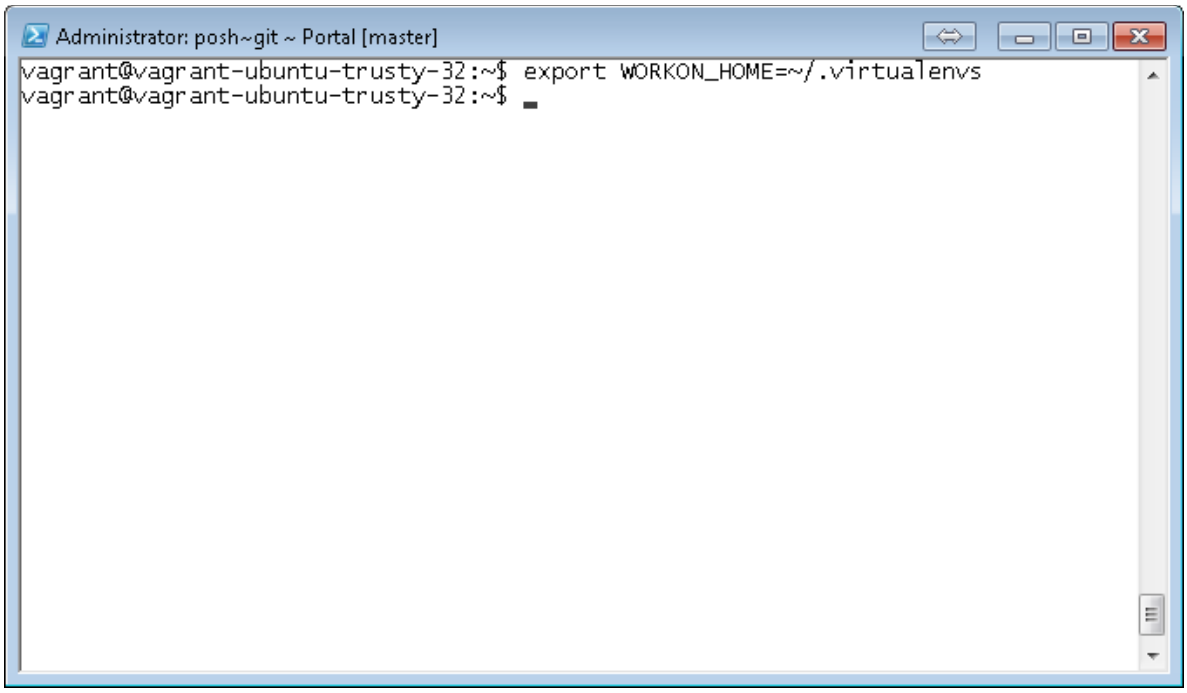
pip3 install --user virtualenvwrapper



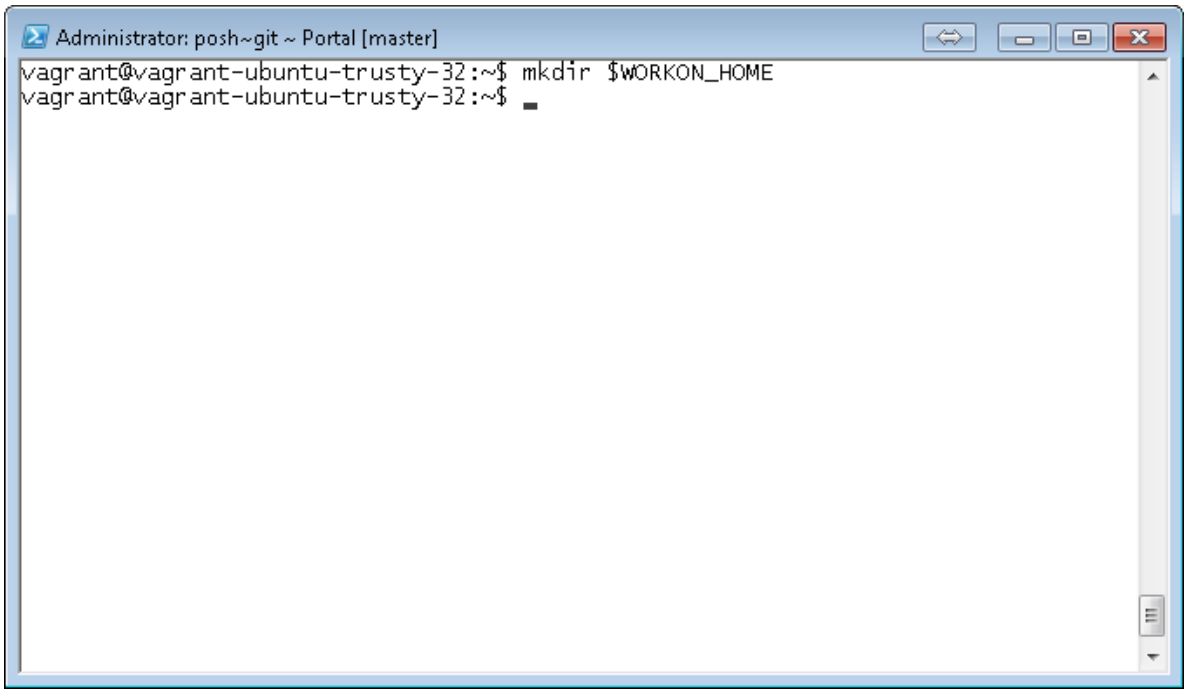
7) Exportar la variable de entorno con

echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3" >> ~/.bashrc

8) Exportar la variable export WORKON_HOME=~/.virtualenvs

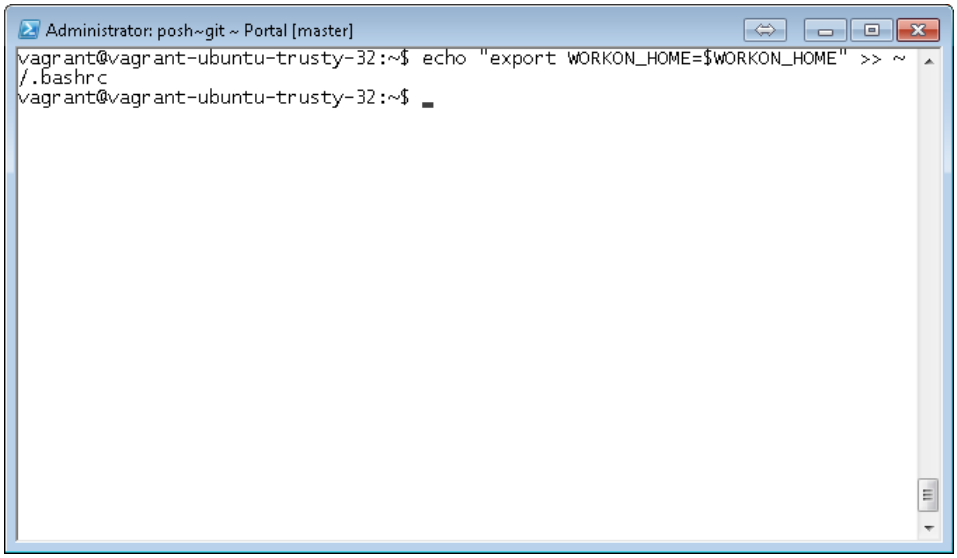


9) Crear el directorio de WORKON_HOME para la distribución

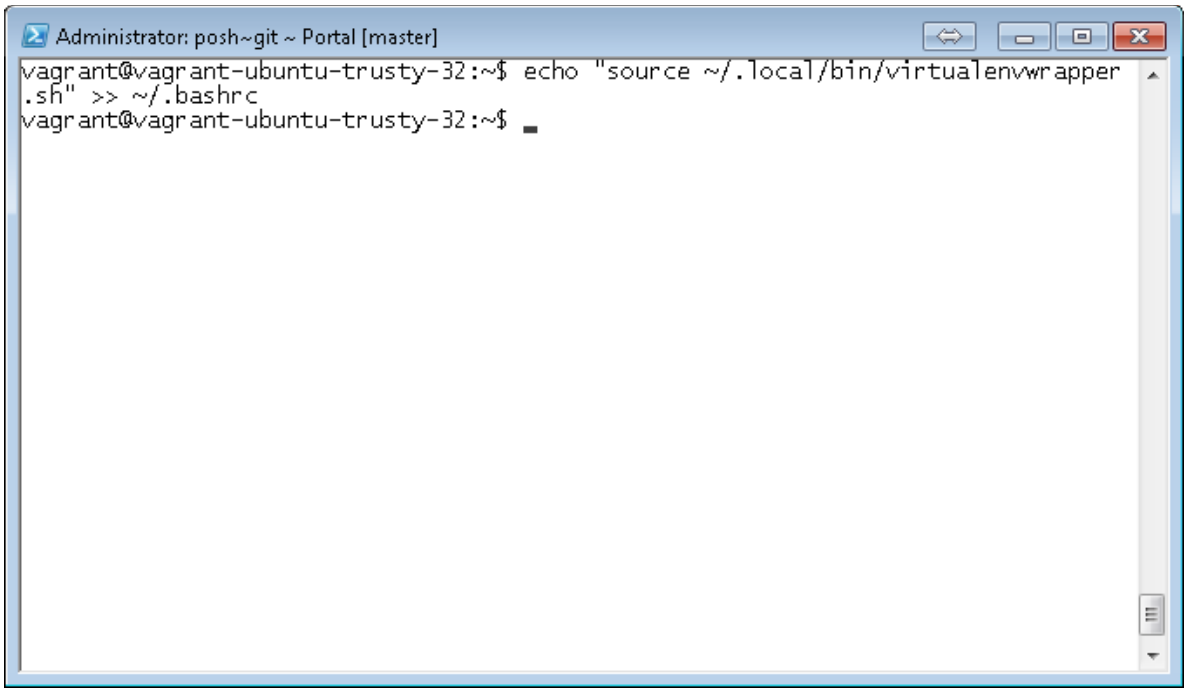


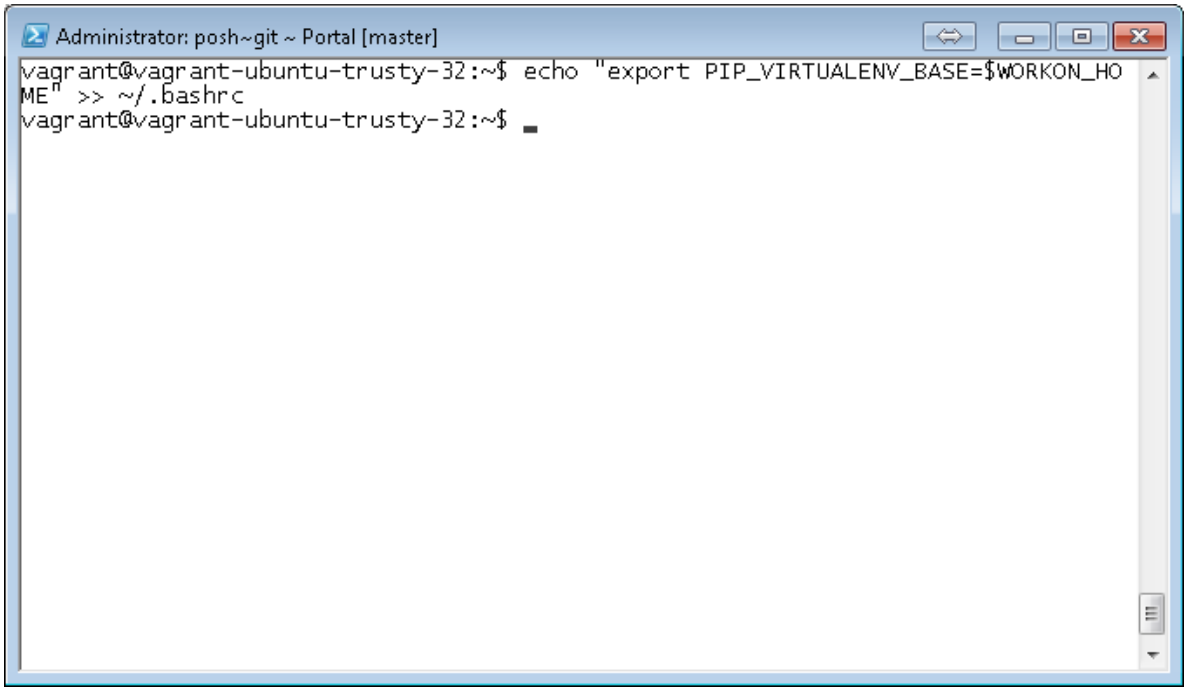
10) Acto seguido se exporta la variable de entorno al Shell, con el comando

echo "export WORKON_HOME=$WORKON_HOME" >> ~/.bashrc



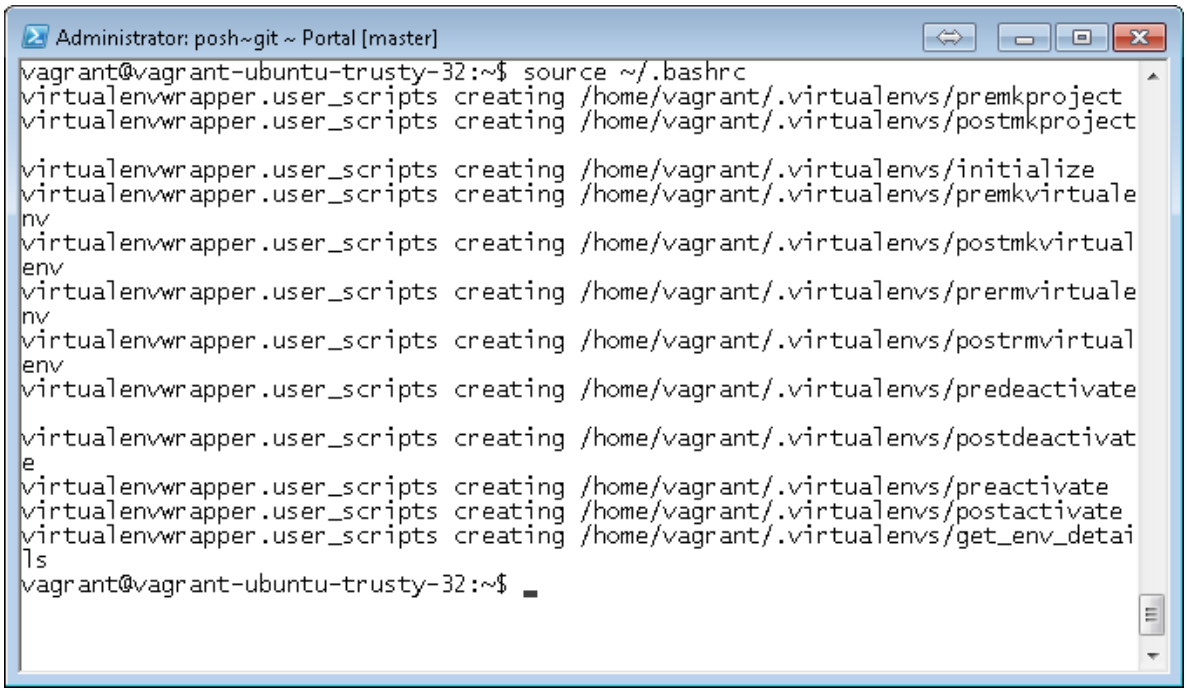
11) Se integra el ambiente de vagrant con virtualenv para crear el ambiente de virtualización para NUVEM CNCFLORA





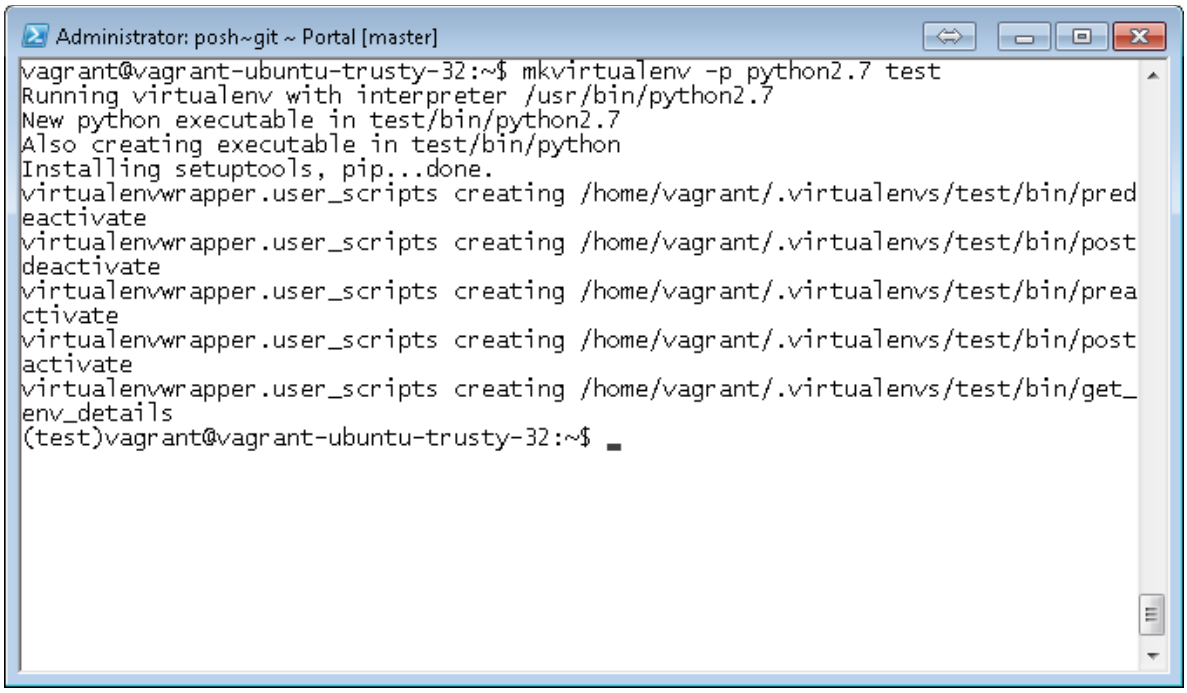
12) Se configura el script de virtualenv para arranque con el Sistema Operativo en los

próximos reinicios.



13) Acto seguido, se procede a validar la instalación con

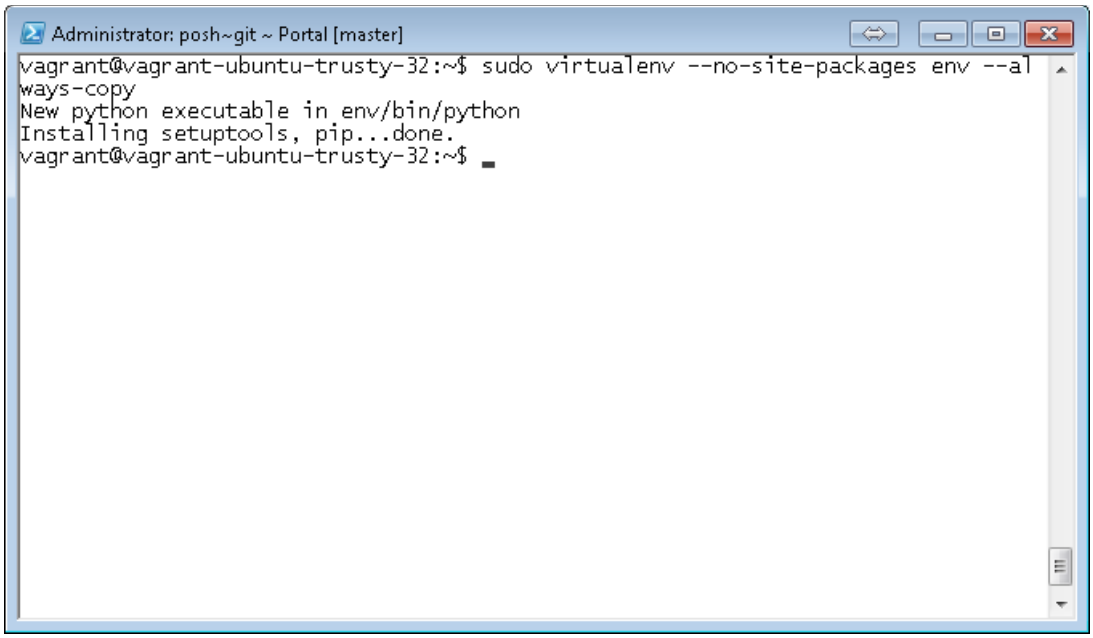
mkvirtualenv -p python2.7 test

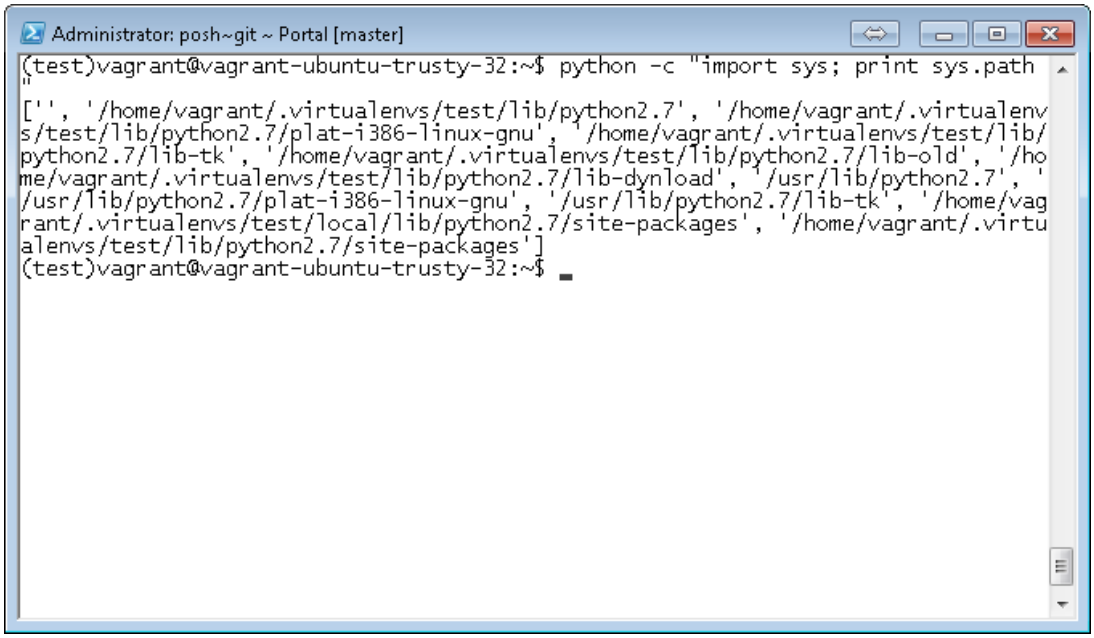


14) Se ejecuta una impresión por consola de PYTHON para validar que el motor (runtime de ejecución), está trabajando correctamente.

Probar con

python -c "import sys; print sys.path"


Administrator: posh~git ~ Portal [master]
vagrant@vagrant-ubuntu-trusty-32:~$ sudo virtualenv --no-site-packages env --al
ways-copy
New python executable in env/bin/python
Installing setuptools, pip...done.
vagrant@vagrant-ubuntu-trusty-32:~$

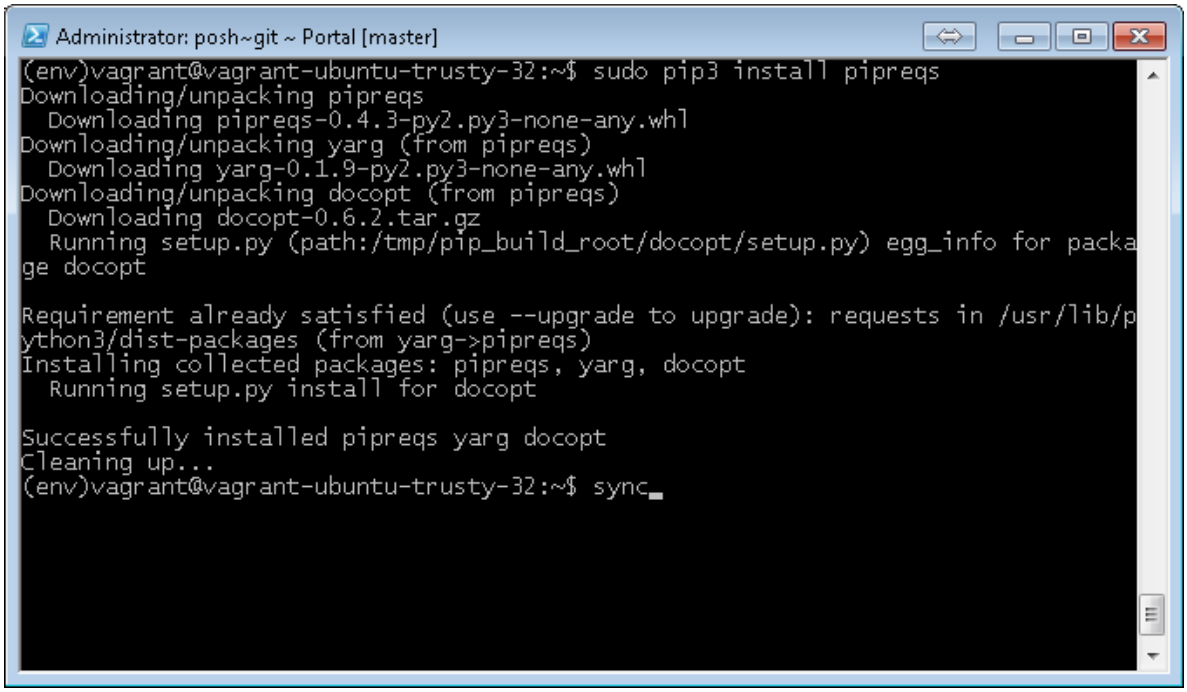

Administrator: posh~git ~ Portal [master]
(test)vagrant@vagrant-ubuntu-trusty-32:~$ python -c "import sys; print sys.path
"
['', '/home/vagrant/.virtualenvs/test/lib/python2.7', '/home/vagrant/.virtualenv
s/test/lib/python2.7/plat-i386-linux-gnu', '/home/vagrant/.virtualenvs/test/lib/
python2.7/lib-tk', '/home/vagrant/.virtualenvs/test/lib/python2.7/lib-old', '/ho
me/vagrant/.virtualenvs/test/lib/python2.7/lib-dynload', '/usr/lib/python2.7', '
/usr/lib/python2.7/plat-i386-linux-gnu', '/usr/lib/python2.7/lib-tk', '/home/vag
rant/.virtualenvs/test/local/lib/python2.7/site-packages', '/home/vagrant/.virtu
alenvs/test/lib/python2.7/site-packages']
(test)vagrant@vagrant-ubuntu-trusty-32:~$

CONFIGURACIÓN DE LIBRERÍAS DE LÍNEA BASE PARA PYTHON

Como recurso del correcto funcionamiento de PYTHON, requerido para el clúster de apps CNCFLORA, se necesitan instalar las librerías adicionales, para lo cual, deben ejecutarse una vez terminado el montaje de línea base Python en la máquina virtual vagrant, los siguientes comandos y para los siguientes propósitos:

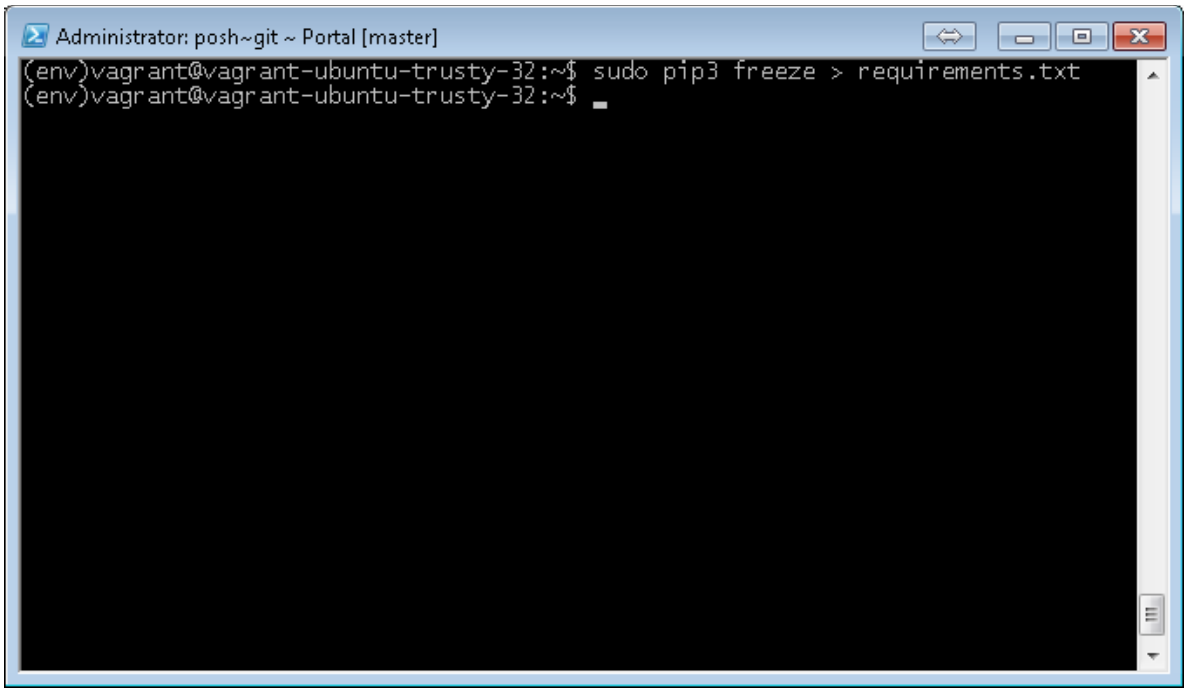
1) Realizar el montaje de pipreqs para Python con el comando:

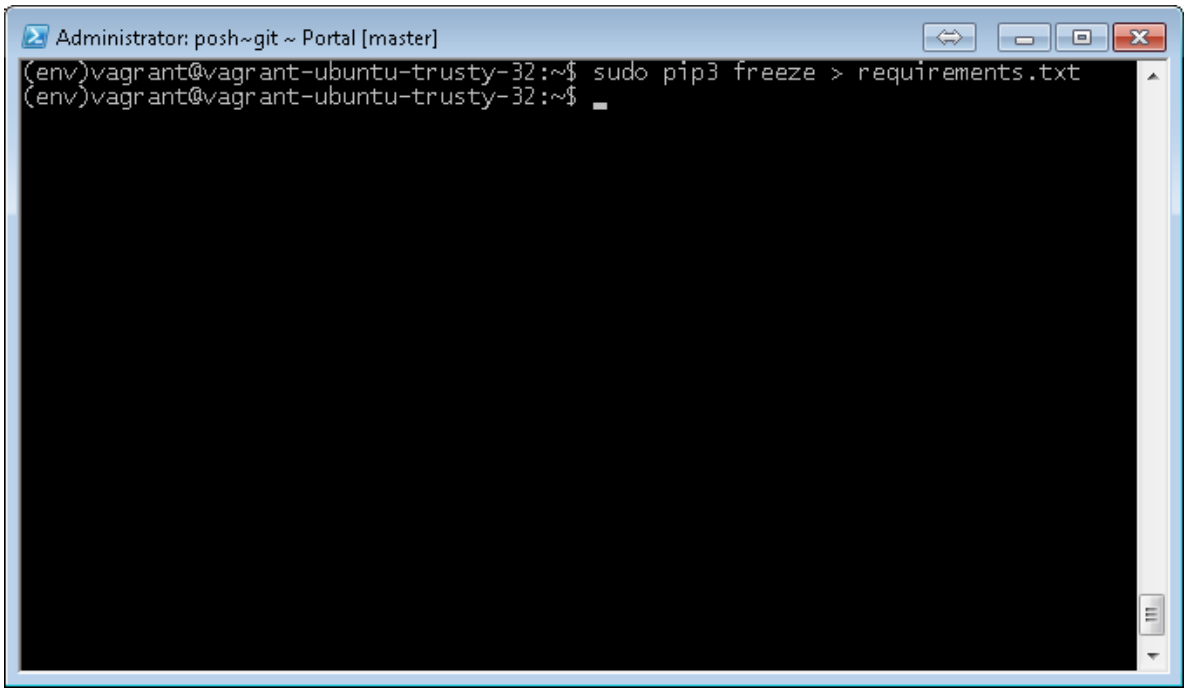
sudo pip install pipreqs

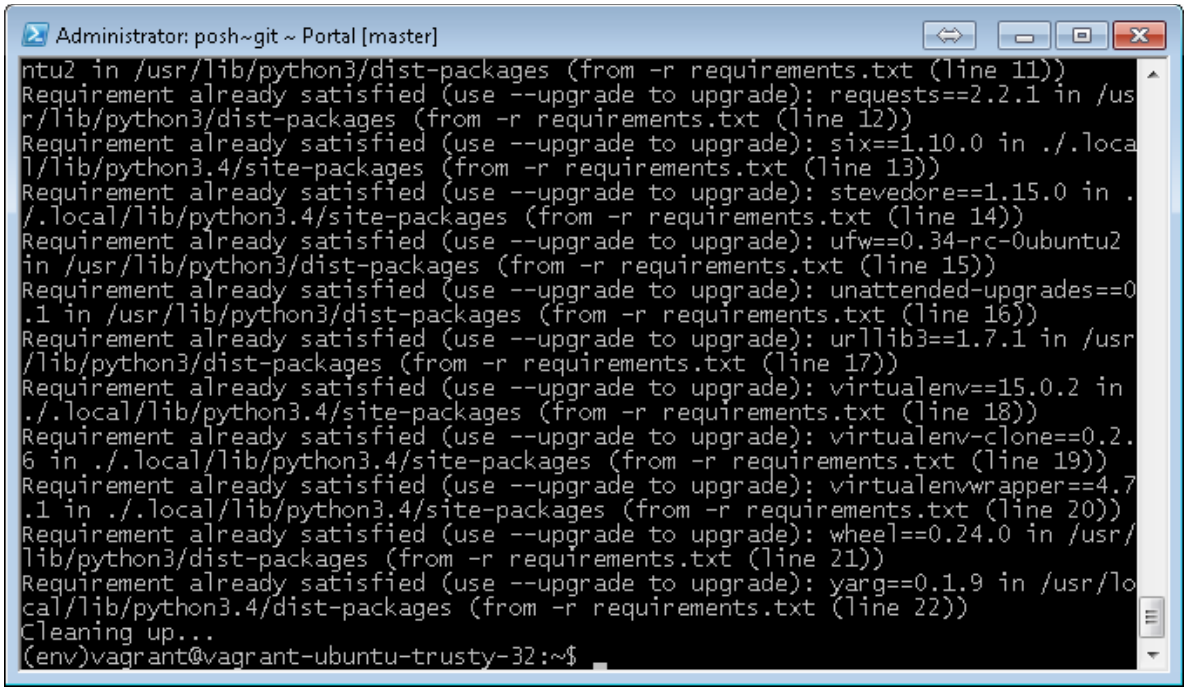


2) Generar backup de los requerimientos de línea base de instalación con el comando:

sudo pip freeze > requirements.txt
sudo pip install -r requirements.txt

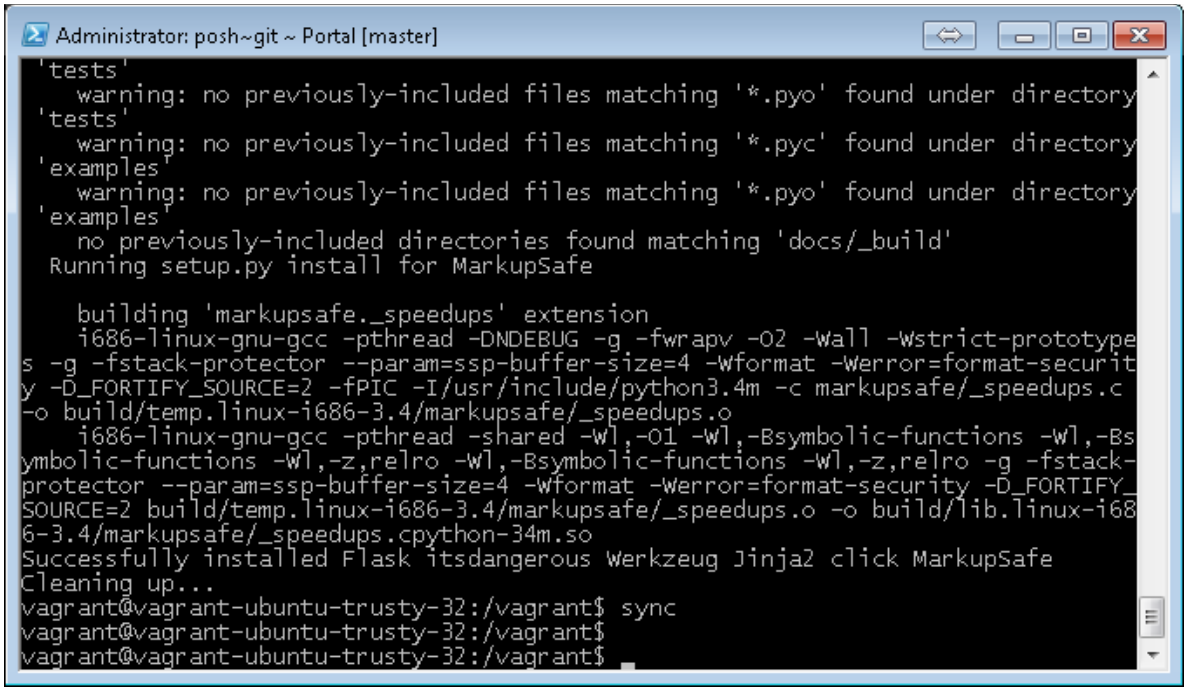

Administrator: posh~git ~ Portal [master]
(env)vagrant@vagrant-ubuntu-trusty-32:~$ sudo pip3 freeze > requirements.txt
(env)vagrant@vagrant-ubuntu-trusty-32:~$


Administrator: posh~git ~ Portal [master]
(env)vagrant@vagrant-ubuntu-trusty-32:~$ sudo pip3 freeze > requirements.txt
(env)vagrant@vagrant-ubuntu-trusty-32:~$



3) Instalar la librería de Flask con el comando

sudo pip3 install Flask

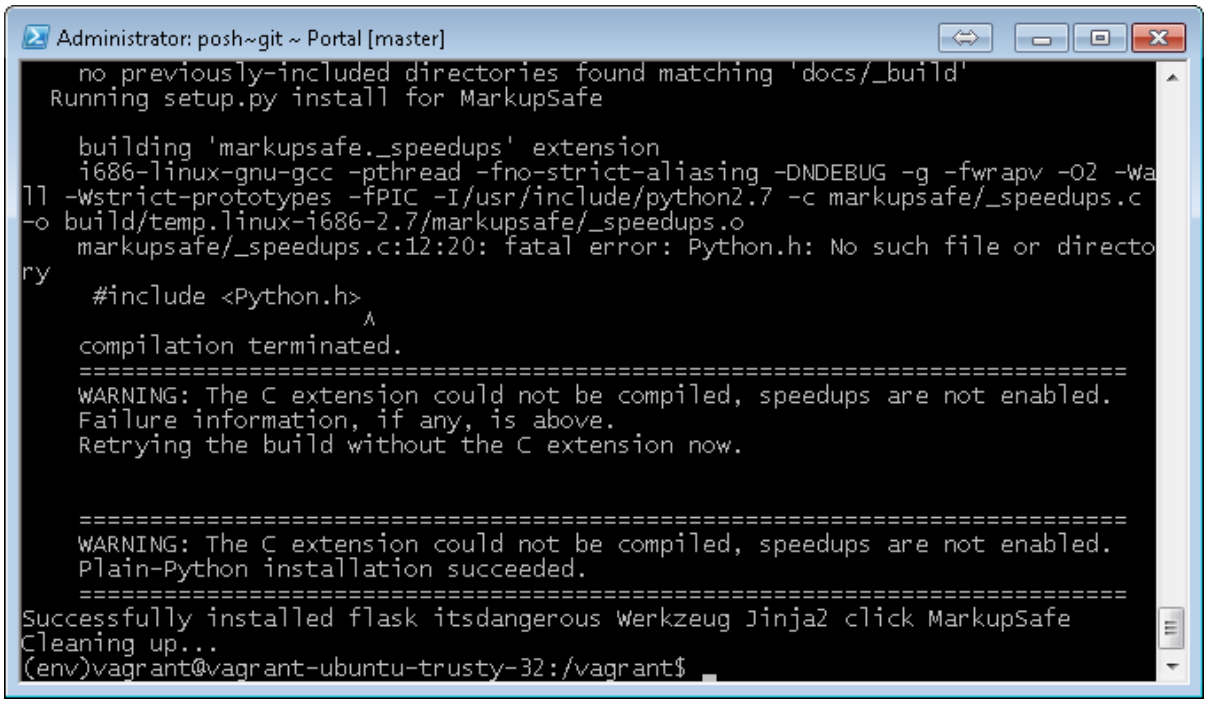


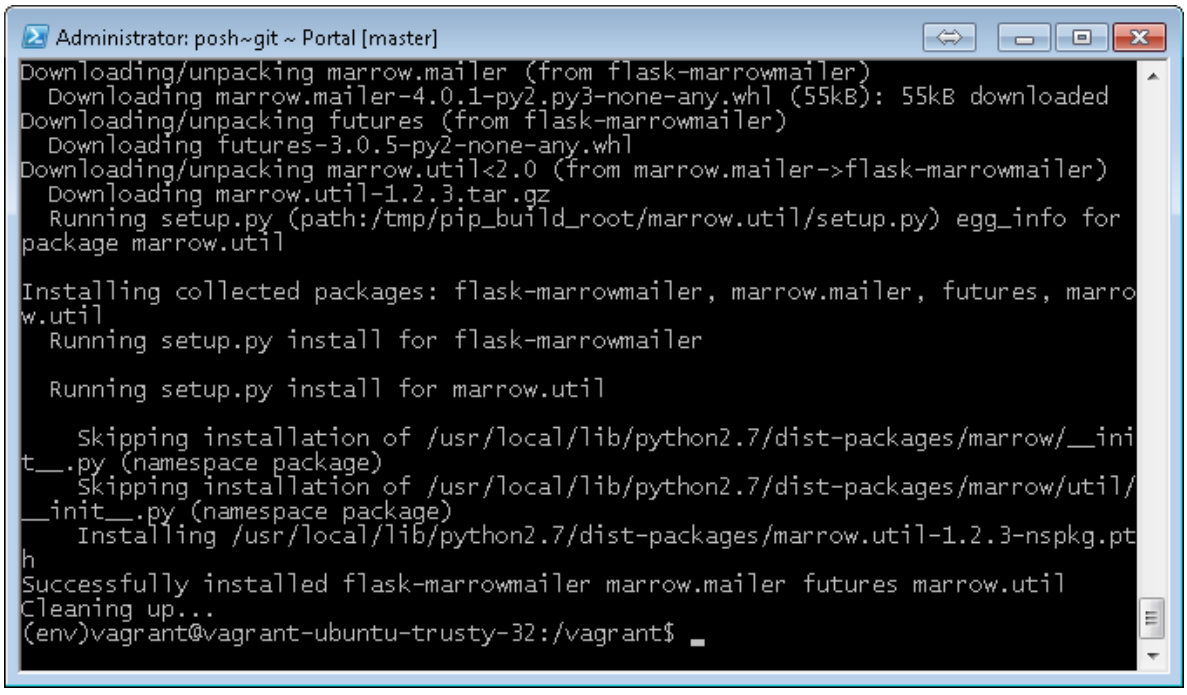
4) Instalar la librería flask (con minúscula) con el comando:

sudo pip install flask

5) Instalar posteriormente la librería de correo flask_marrowmailer, con el comando

sudo pip install flask_marrowmailer





6) Posteriormente se debe instalar la librería netifaces, requerida para la resolución en Python de la dirección local de la máquina (como obtención de la dirección IP del servidor de resolución) y de esta manera, poder configurar rutas de acceso al contexto o configurar hipervínculos de forma dinámico en

código PYTHON/HTML. Se debe proceder con el conjunto de comandos

sudo apt-get install python-dev

sudo pip install netifaces

5. ACTUALIZACIÓN DE APACHE PARA LA CONFIGURACIÓN DEL PROXY DEL SERVIDOR

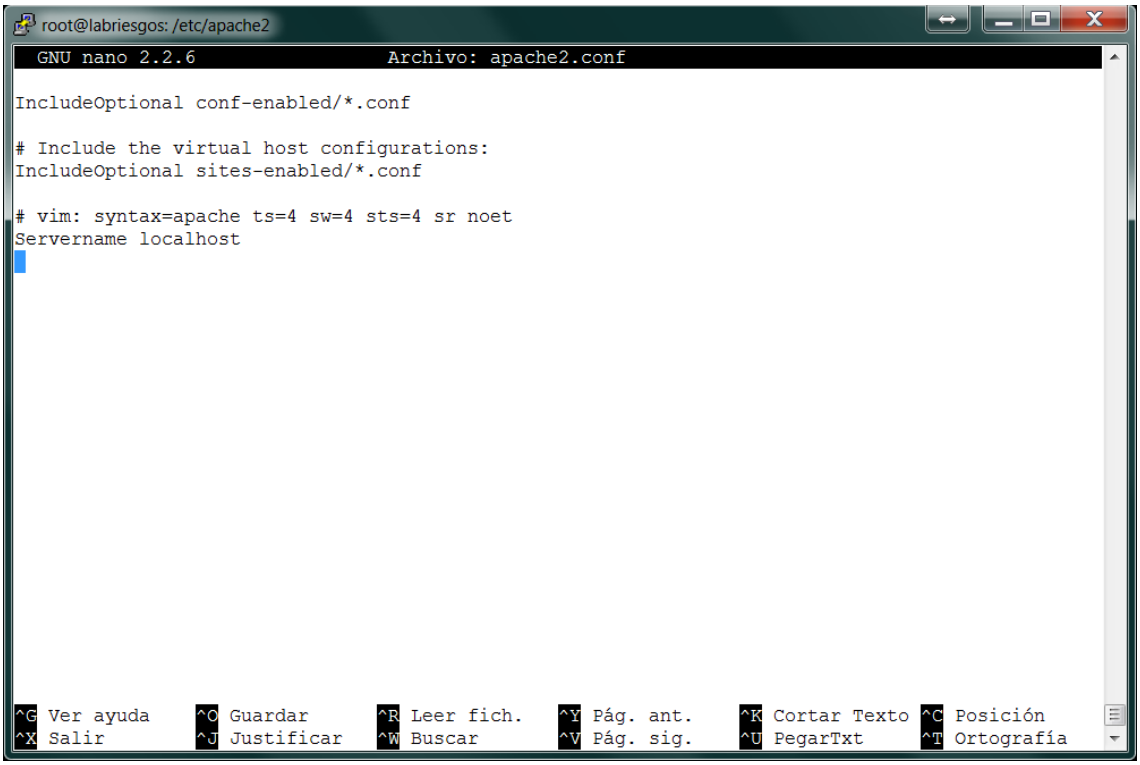
Con el fin de evitar el error BAD GATEWAY 502, se despliega luego de colocar a correr el clúster de aplicaciones APP NUVEM CNCFLORA, se debe instalar Apache con las siguientes instrucciones:

a) sudo apt-get update
b) sudo apt-get install apache2
c) sudo apt-get install libapache2-mod-php5 php5 php5-mcrypt

Acto seguido, se debe configurar el archivo apache2.conf, con la siguiente línea

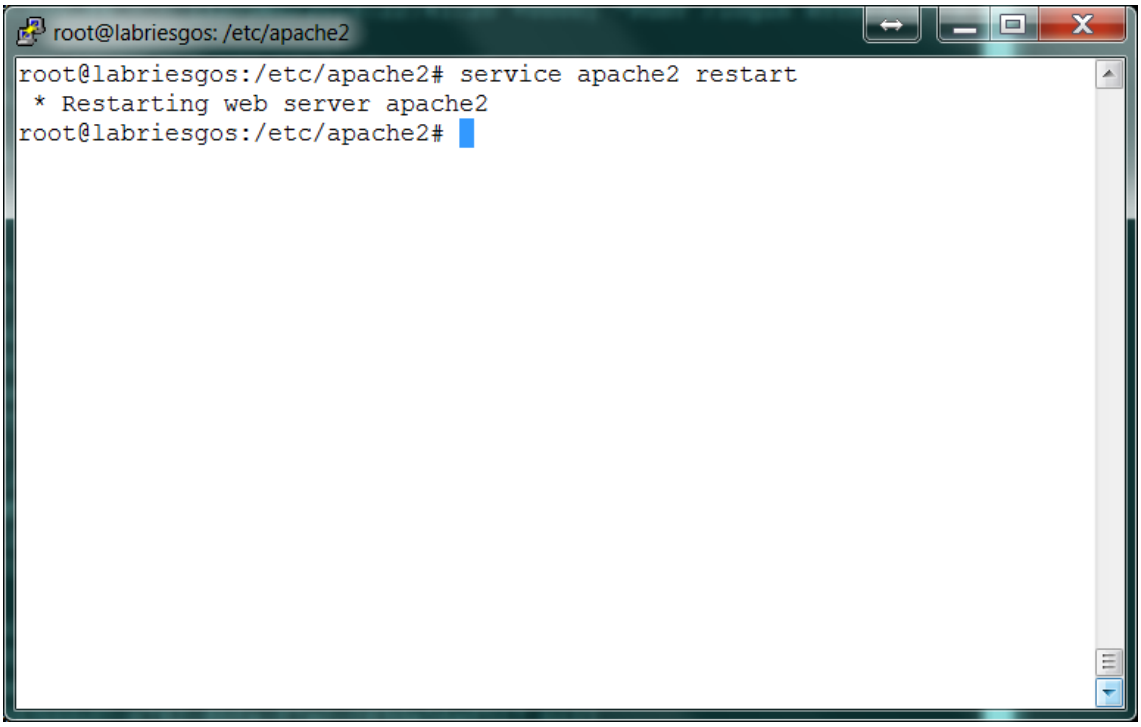
ServerName localhost, tal como se muestra a continuación:

/etc/apache2/apache2.conf

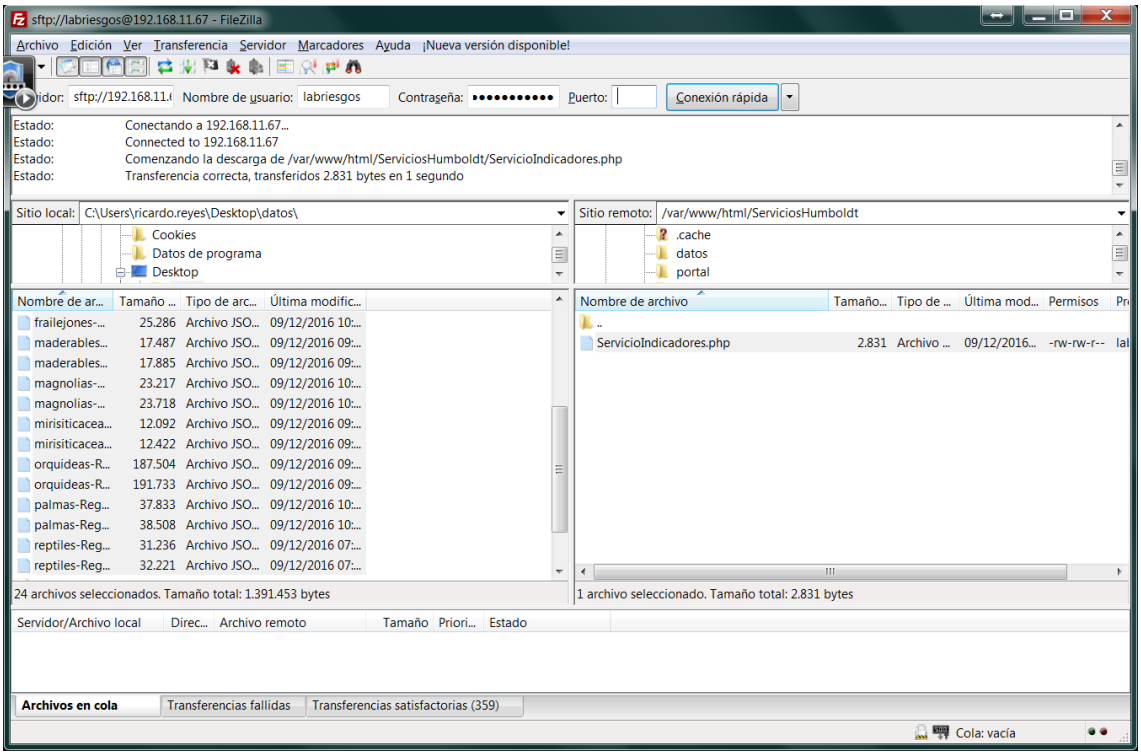


Y reiniciar el servicio de apache con

service apache2 restart

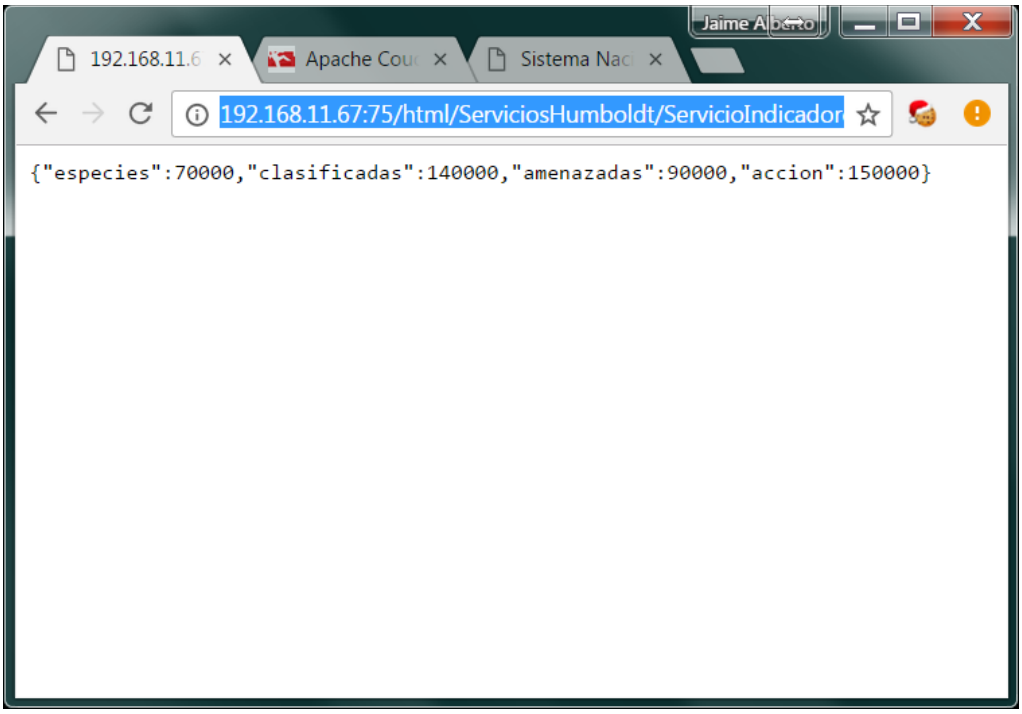


Posteriormente validar la instalación /var/www/html en la raíz del disco y desplegar los Servicios Web PHP desarrollados que hablarán con los handlers Python del App de Portal de CNCFLORA, tal como se ilustra a continuación:



Resolver la URI con

http://192.168.11.67:75/html/ServiciosHumboldt/ServicioIndicadores.php



Validar la salida JSON y configurar la URI en el consumo del componente Python como se muestra a continuación:

```
ni.ifaddresses('eth1')
direccionServidor = ni.ifaddresses('eth1')[2][0]['addr']
uriServicio = 'http://' + direccionServidor +
':75/html/ServiciosHumboldt/ServicioIndicadores.php'
resp = requests.get(uriServicio)
if resp.status_code != 200:
   raise ApiError('GET /tasks/ {}'.format(resp.status_code))

datos = json.loads(resp.text)
especies = datos['especies']
clasificadas = datos['clasificadas']
amenazadas = datos['amenazadas']
accion = datos['accion']
#*****************************************************************
#*****************************************************************
#*****************************************************************
return  render_template('home.html',base_url=self.base_url,static_
```

6. INSTALACIÓN DE DOCKER PARA EL CLÚSTER DE APLICACIONES APP NUVEM CNCFLORA

Para instalar el clúster se deben ejecutar los siguientes comandos en la MV instalada:

1) sudo apt-get update

2) sudo apt-get install apt-transport-https ca-certificates

3) Montar el repositorio con:

sudo apt-key adv --keyserver hkp://pgp.mit.edu:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D

4) Actualizar el archivo del repo con:

sudo echo 'deb https://apt.dockerproject.org/repo ubuntu-trusty main' > /etc/apt/sources.list.d/docker.list

5) sudo apt-get update

6) sudo apt-get install docker-engine –y

7) sudo service docker start

8) sudo curl -L https://github.com/docker/compose/releases/download/1.6.2/docker-compose-`uname -s`-`uname -m` >$sudo chmod +x /usr/local/bin/docker-compose

9) sudo usermod -aG docker vagrant

10) sudo apt-get install docker

11) sudo pip install docker-compose

12) sudo docker-compose up –d (Descargar todos los paquetes y esperar que inicien los servicios)

13) Una vez se validen que los servicios suben, editar el docker-compose.yml, con la siguiente estructura:

```
version: "2"
services:
  proxy:
    image: diogok/caddy
    restart: "always"
    ports:
      - 80:80
      - 443:443
    networks:
      - nuvem
    volumes:
      - ./Caddyfile:/etc/Caddyfile
      - ./index.html:/var/www/html/index.html
      - /opt/caddy:/root/.caddy
  couchdb:
    image: cncflora/couchdb
    networks:
      - nuvem
    restart: "always"
    ports:
       - "5984:5984"
    volumes:
       - "/var/data/couchdb:/var/lib/couchdb:rw"
  elasticsearch:
    image: cncflora/elasticsearch
```

```
    networks:
      - nuvem
    restart: "always"
    ports:
       - "9200:9200"
    volumes:
       - "/var/data/elasticsearch/data:/usr/share/elasticsearch/data:rw"
kibana:
    image: cncflora/kibana
    networks:
      - nuvem
    restart: "always"
floradata:
    image: cncflora/floradata
    networks:
      - nuvem
    restart: "always"
    ports:
       - 80
       - 9001
aka:
    image: cncflora/aka
    networks:
      - nuvem
    restart: "always"
```

```
    volumes:
      - "/var/data/aka:/var/data/aka:rw"
    environment:
      PROXY: "/aka"
services:
    image: cncflora/services
    networks:
     - nuvem
    restart: "always"
    environment:
      DB: "public"
      BASE: "/services"
#portal:
#    image: cncflora/portal
#    networks:
#     - nuvem
#    restart: "always"
#    environment:
#      BASE: "/portal"
entrypoint:
    image: cncflora/entrypoint
    networks:
     - nuvem
    restart: "always"
connect:
```

```
    image: cncflora/connect
    networks:
      - nuvem
    restart: "always"
    volumes:
       - "/var/data/connect:/var/floraconnect:rw"
    environment:
       PROXY: "/connect"
dwc_services:
    image: cncflora/dwc-services
    networks:
      - nuvem
    restart: "always"
    environment:
       PROXY: "/dwc_services"
checklist:
    image: cncflora/checklist
    networks:
      - nuvem
    restart: "always"
    environment:
       BASE: "/checklist"
       HOST: "${HOST}"
occurrences:
    image: cncflora/occurrences
```

```
    networks:
      - nuvem
    restart: "always"
    environment:
       BASE: "/occurrences"
       HOST: "${HOST}"
profiles:
    image: cncflora/profiles
    networks:
      - nuvem
    restart: "always"
    environment:
       BASE: "/profiles"
       HOST: "${HOST}"
assessments:
    image: cncflora/assessments
    networks:
      - nuvem
    restart: "always"
    environment:
       BASE: "/assessment"
       HOST: "${HOST}"
migrator:
    image: cncflora/migrator
    networks:
```

```
      - nuvem
    restart: "always"
    environment:
      HOST: "${HOST}"
reports2:
    image: cncflora/reports2
    networks:
      - nuvem
    restart: "always"
arquivos:
    image: cncflora/apache
    networks:
      - nuvem
    restart: "always"
    volumes:
      - "/var/data/www:/var/www:rw"
arquivos-sftp:
    image: asavartzeth/sftp
    networks:
      - nuvem
    restart: "always"
    ports:
      - "2223:22"
    environment:
      SFTP_USER: "cncflora"
```

```
      SFTP_PASS: "1cncflora2"
    volumes:
      - "/var/data/www:$SFTP_DATA_DIR/chroot/www:rw"
networks:
 nuvem:
```

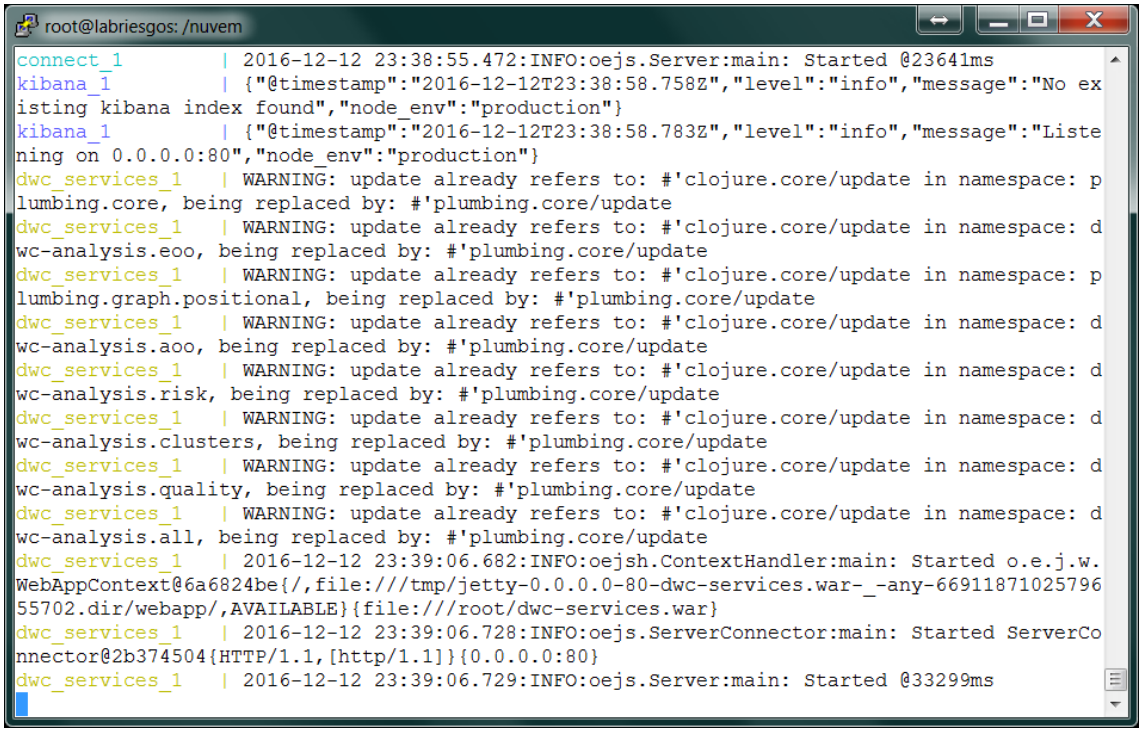
Solo comentariar la sesión de Portal y arrancar el comando con:

export HOST=IP_SERVIDOR_AMBIENTE

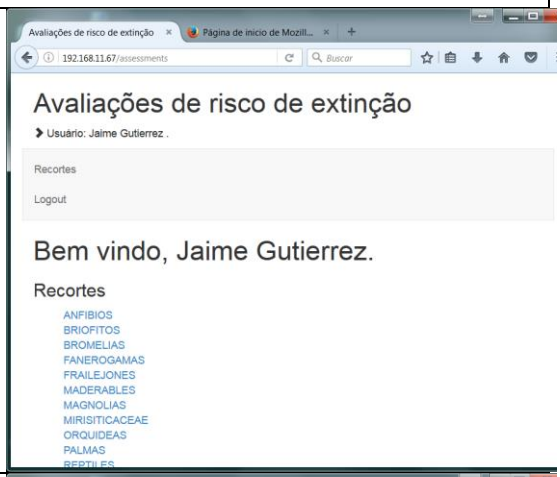
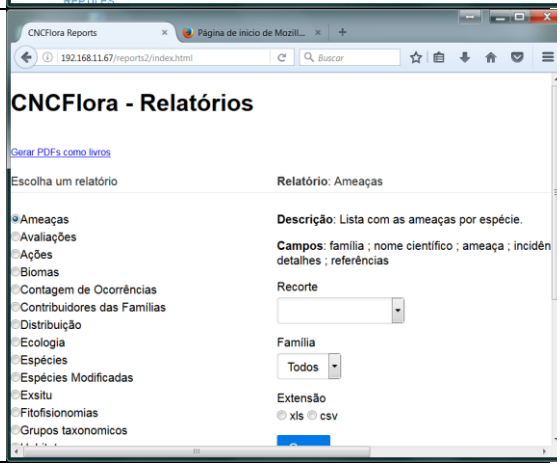
ejemplo

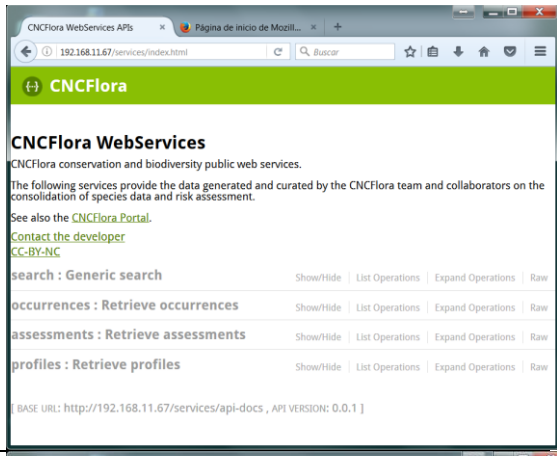
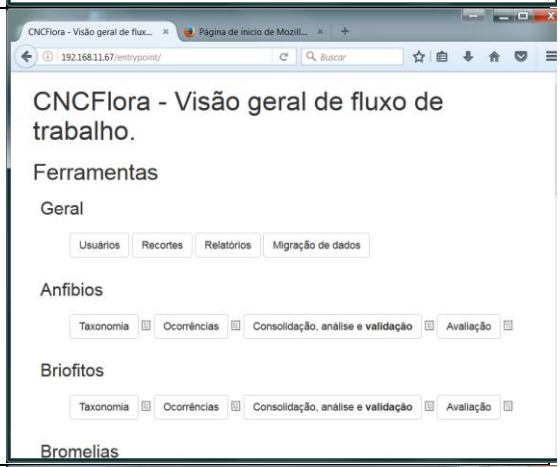
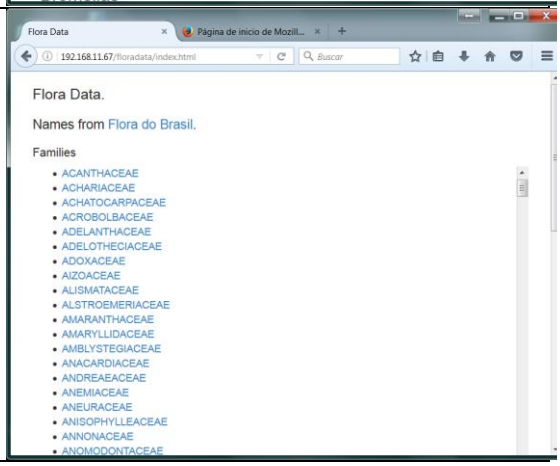
export HOST=192.168.11.67, y luego

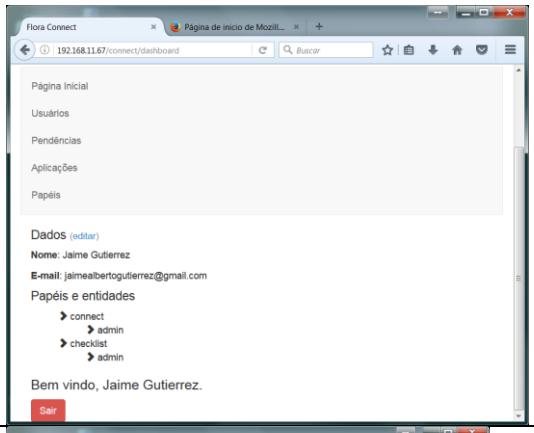
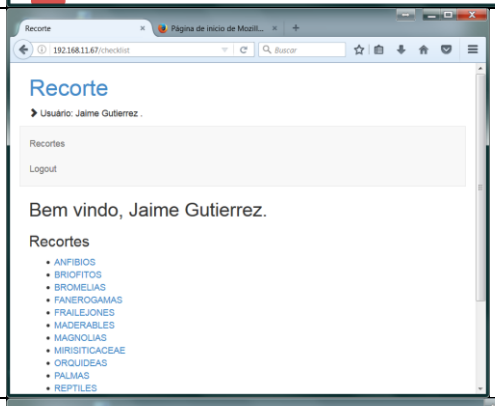
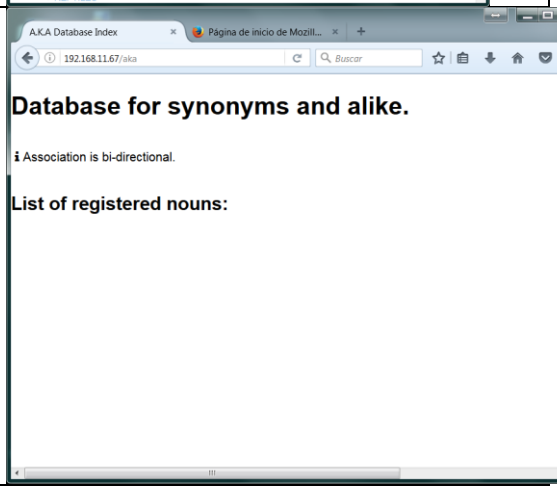
sudo docker-compose up en la carpeta **/vagrant**

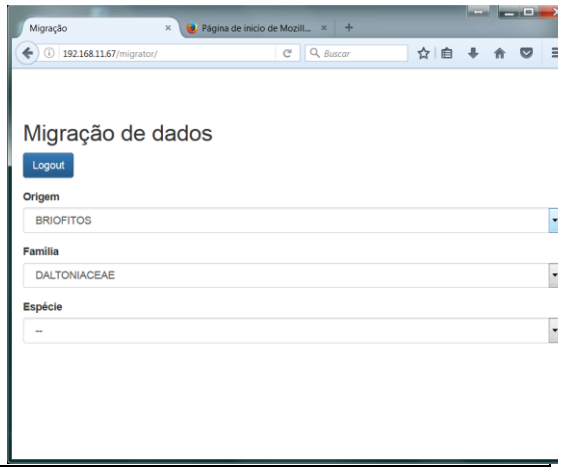


Validar el acceso a aplicaciones cluster, en las siguientes URI:

Aplicación	URI de acceso	APLICACIÓN
Occurrences	**http://192.168.11.67/occurrences/**	
Assessment	**http://192.168.11.67/assessments**	
reports	**http://192.168.11.67/reports2/index.html**	

Services	**http://192.168.11.67/services/index.html**	
entrypoint	http://192.168.11.67/entrypoint/	
floradata	**http://192.168.11.67/floradata/index.html**	

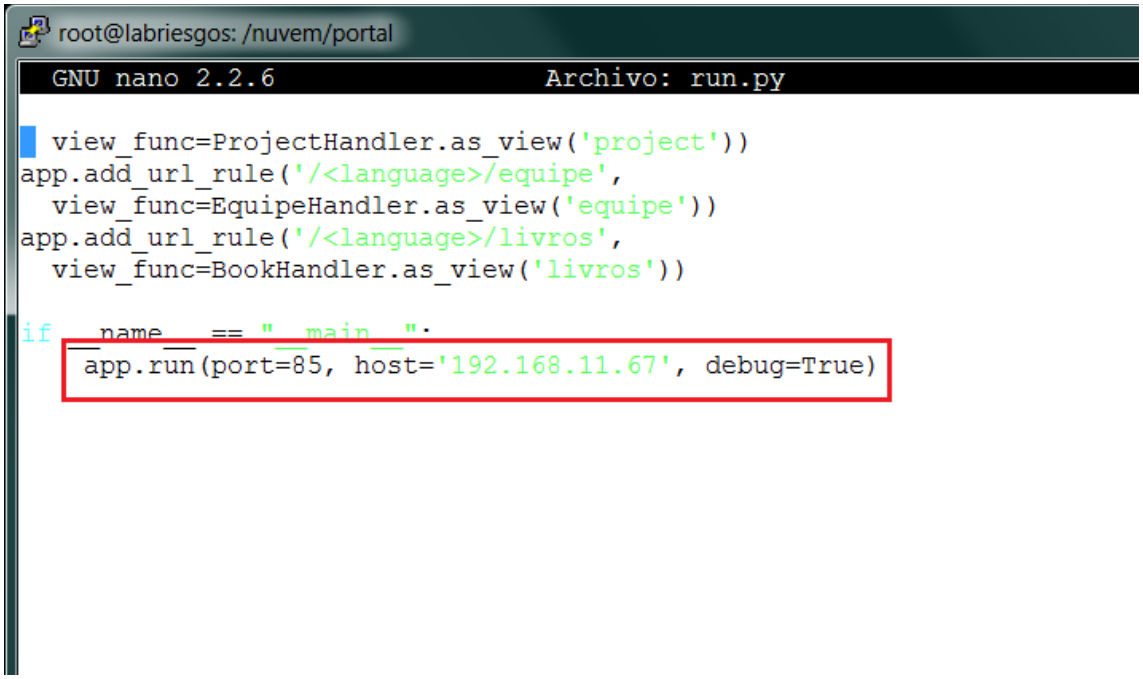
Connect	**http://192.168.11.67/connect/index**	
checklist	**http://192.168.11.67/checklist**	
Aka	**http://192.168.11.67/aka**	

migrator	**http://192.168.11.67/migrator/**	

7. ARRANQUE DEL AMBIENTE DE FRONT END PORTAL NUVEM CNCFLORA

1) Ingresar en la consola luego de sesionar a la carpeta /vagrant/portal

2) Editar el archivo run.py y establecer la configuración de depuración y la dirección IP del servidor



3) Crear el comando arranque.sh, con el siguiente contenido:

```
echo "****************************************"
echo "****************************************"
sudo python run.py
echo "****************************************"
echo "****************************************"
```

4) Asignar permiso de ejecución al script con

chmod +777 arranque.sh

5) Ejecutar y validar el arranque del servidor

7) Resolver el Portal en la URI

http://192.168.11.67:85/

Elaboró:
Ingeniero
JAIME ALBERTO GUTIÉRREZ MEJÍA
CONSTRATISTA INSTITUTO ALEXANDER VON HUMBOLDT
DICIEMBRE 2016