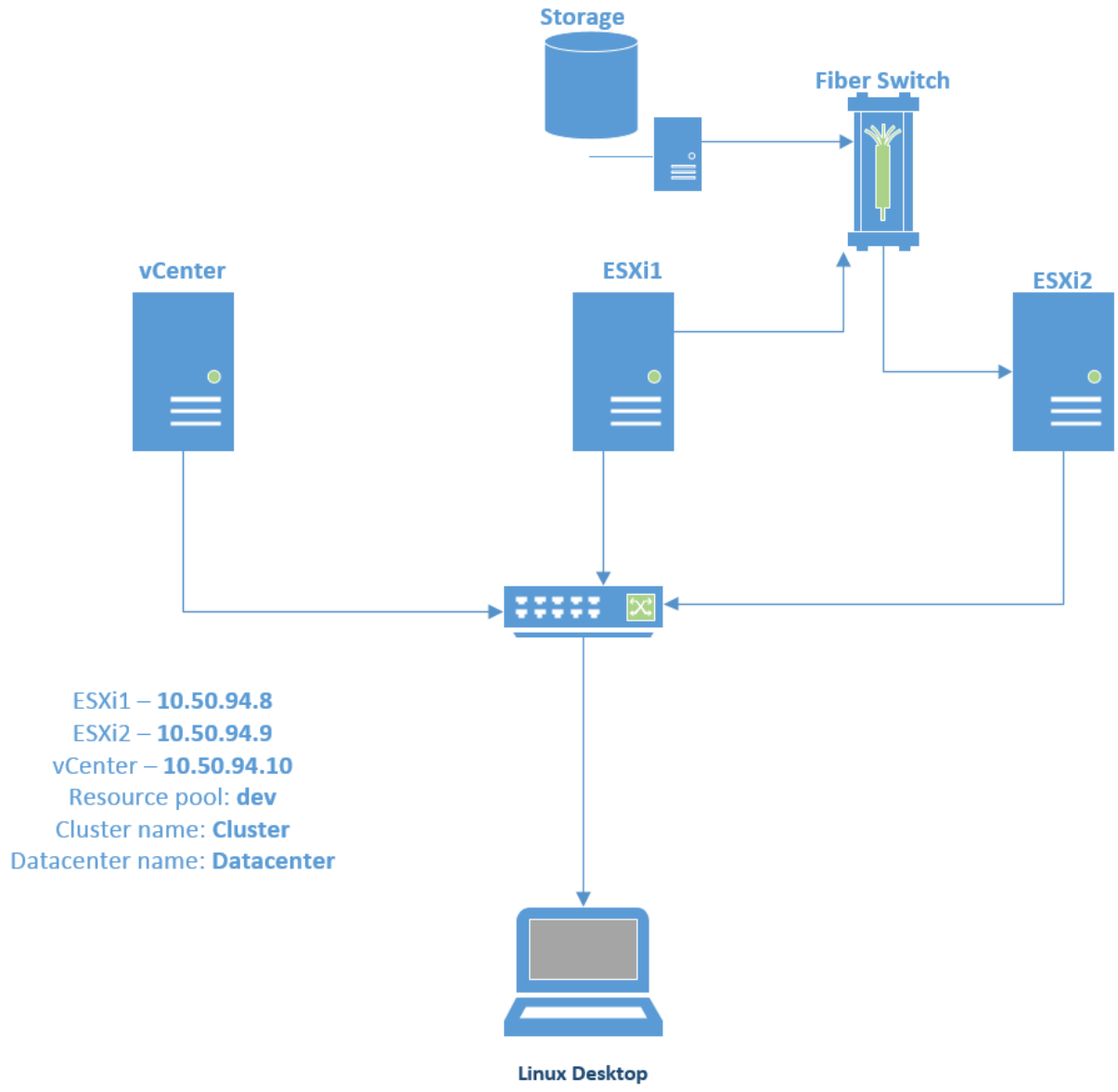
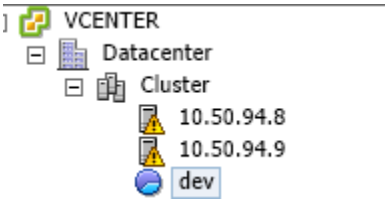
Our goal to configure virtual machines automatically via VMware vSphere API. We will use officially API's of VMWare **pyvmomi** and third party library of **ezmomi** which is used pyvmomi library.

We have 3 physical servers. One of them vCenter server and 2 ESXi's. ESXi servers already connected to the shared storage and works in the cluster. Our datacenter name is **"Datacenter"**, cluster name is **"Cluster"** and resource pool name is **"dev"**.

VMWare topology as following:





We must prepare our desktop for use libraries. I used Fedora24 desktop. To use Python codes we must install **Python27, pip** to install python libraries and **git** to download codes.

```
# dnf install python2
# dnf install git sshpass
# python -m ensurepip
# python -m pip install pyvmomi
# python -m pip install ezmomi
```

Clone codes from repository:

```
$ git clone https://github.com/sijis/pyvmomi-examples
```

Go to downloaded folder and look at the new files:

```
$ cd pyvmomi-examples/
[jshahverdiev@vagrantdesktop pyvmomi-examples]$ ls -la
drwxrwxr-x   3 jshahverdiev jshahverdiev 4096 Dec 29 17:39 .
drwx------. 30 jshahverdiev jshahverdiev 4096 Dec 29 17:39 ..
-rw-rw-r--   1 jshahverdiev jshahverdiev 2319 Dec 29 17:39 create-vm.py
drwxrwxr-x   8 jshahverdiev jshahverdiev 4096 Dec 29 17:39 .git
-rw-rw-r--   1 jshahverdiev jshahverdiev    6 Dec 29 17:39 .gitignore
-rw-rw-r--   1 jshahverdiev jshahverdiev  925 Dec 29 17:39 migrate-vm.py
-rw-rw-r--   1 jshahverdiev jshahverdiev  227 Dec 29 17:39 README.rst
-rw-rw-r--   1 jshahverdiev jshahverdiev  615 Dec 29 17:39 reboot-vm.py
-rw-rw-r--   1 jshahverdiev jshahverdiev 2680 Dec 29 17:39 vmutils.py
```

I will show some lines of codes which I changed. By default it is trying to connect to the vcenter server via SSL. I turned off this because I use it in test environment(But its not good for security reason).

Changed lines in the **create-vm.py** file:

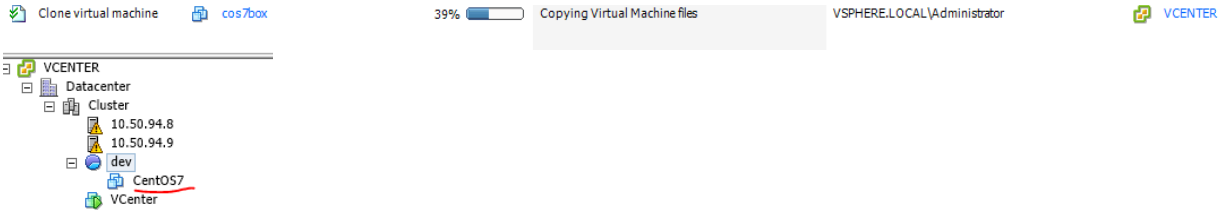
```
# Added two libraries to turn off SSL and requests warnings.
import requests
import ssl

# Disabling urllib3 ssl warnings
requests.packages.urllib3.disable_warnings()

# Disabling SSL certificate verification
context = ssl.SSLContext(ssl.PROTOCOL_TLSv1)
context.verify_mode = ssl.CERT_NONE

# Added last argument to the SmartConnect function:
try:
    si = SmartConnect(host=vcenter, user=username, pwd=password, port=443,
sslContext=context)
except IOError, e:
    pass

# Added the name of template file of virtual machine
template_vm = vmutils.get_vm_by_name(si, 'cos7box')
```

```
# Changed resource pool name to name which we created before 'dev'
resource_pool = vmutils.get_resource_pool(si, 'dev')
```

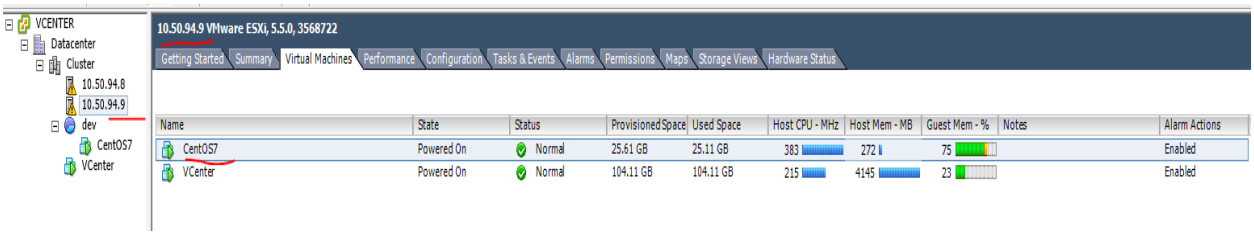
Create new virtual machine from Linux console:

```
[jshahverdiev@vagrantdesktop pyvmomi-examples]$ python create-vm.py
Username: administrator@vsphere.local
Password: pass_for_vcenter_admin_user
vcenter: 10.50.94.10
New vm name: CentOS7
```

At the same time look at the vsphere client console:



If you want to migrate your VM from one ESXI node to other we must use the **migrate.py** script. In the following screenshot as we see our CentOS7 virtual machine works in the 10.50.94.9 ESXi machine. Don't forget you must configure vmotion for this.



To disable SSL, I added the same lines which was in the **create-vm.py** file and added **sslContext=context** argument to the **SmartConnect** function:

```
[jshahverdiev@vagrantdesktop pyvmomi-examples]$ python migrate-vm.py
Username: administrator@vsphere.local
Password:
vcenter: 10.50.94.10
VM: CentOS7
ESX Host: 10.50.94.8
```

Reboot the virtual machine. Added same lines to file which told before and added function argument to the **SmartConnect**:

```
[jshahverdiev@vagrantdesktop pyvmomi-examples]$ python reboot-vm.py
Username: administrator@vsphere.local
Password:
vcenter: 10.50.94.10
VM: CentOS7
```

Using ezmomi library

Ezmomi is very comfortable library. With this library we can use ezmomi command via CLI. As we already installed ezmomi from PIP let start to use it.

Look at help and to see which possibilities we can use:

```
[jshahverdiev@vagrantdesktop pyvmomi-examples]$ ezmomi --help
usage: ezmomi [-h] [--version] [--debug]

{list,listSnapshots,createSnapshot,removeSnapshot,revertSnapshot,clone,destroy,status,shutdown,po
werOff,powerOn,syncTimeWithHost}
             ...

Perform common vSphere API tasks

positional arguments:

{list,listSnapshots,createSnapshot,removeSnapshot,revertSnapshot,clone,destroy,status,shutdown,po
werOff,powerOn,syncTimeWithHost}
                        Command
    list                List VMware objects on your VMware server
    listSnapshots       List snapshots for a VM
    createSnapshot      Create snapshot for a VM
    removeSnapshot      Remove snapshot for a VM
    revertSnapshot      Revert snapshot for a VM
    clone               Clone a VM template to a new VM
    destroy             Destroy/delete a Virtual Machine
    status              Get a Virtual Machine's power status
    shutdown            Shutdown a Virtual Machine (will fall back to powerOff
                        if guest tools are not running)
    powerOff            Power Off a Virtual Machine (not a clean shutdown)
    powerOn             Power On a Virtual Machine
    syncTimeWithHost    Virtual Machine syncs time with host

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
  --debug               Print debug messages
```

To use ezmomi firstly we must configure **config.yml** file to define connection credentials.

Create configuration file and edit creadentials for your needs(user, Pass, Host, Template file, Network for VMs):

```
$ mkdir ~/.config/ezmomi/ && vim ~/.config/ezmomi/config.yml
$ cat ~/.config/ezmomi/config.yml
# config.yml
# ezmomi

# Mail setup for notifications
mail: true
mailserver: localhost
mailfrom: jamal.shahverdiev@example.com
mailto: jamal.shahverdiev@example.com

# vCenter connection settings
```

```
server: 10.50.94.10
port: 443
username: administrator@vsphere.local
password: "PassforVCenter"

# New VM defaults
cpus: 1
mem: 3
template: cos7box
domain: example

dns_servers: ['8.8.8.8','8.8.4.4']

# List your networks here
networks:
  '10.50.94.0/24':
    datacenter: 'Datacenter'
    cluster:    'Cluster'
    datastore:  'Lun2'
    network:    'VM Network'
    gateway:    '10.50.94.1'

#  '192.168.24.50/24':
#    datacenter: 'Foocenter'
#    cluster:    'FooCluster'
#    datastore:  'Foomount'
#    network:    'Internal 2'
#    gateway:    '192.168.24.50'

  # storage network
  # Note: any networks that not used for primary IPs only need to have the
network name defined, like this example.
#  '172.20.200.1/24':
#    network:    'Database Storage'
```

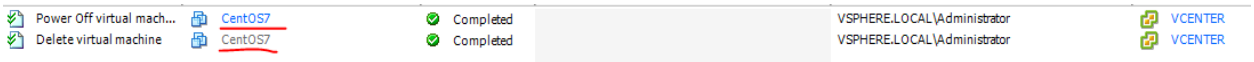
If you want to disable requests warnings then just add the following lines to the **/usr/bin/ezmomi** file:

```
import requests
requests.packages.urllib3.disable_warnings()
```

Now try to destroy **CentOS7** virtual machine:

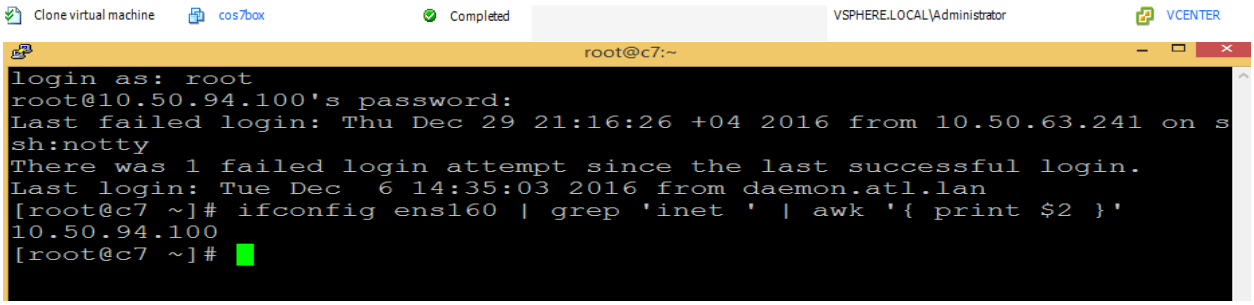
```
[jshahverdiev@vagrantdesktop pyvmomi-examples]$ echo yes | ezmomi destroy --
name CentOS7
Do you really want to destroy CentOS7 ? [yes/no] yes
Finding VirtualMachine named CentOS7...
Found VirtualMachine: 'vim.VirtualMachine:vm-70' Name: CentOS7
Destroying CentOS7...
```

As we see virtual machine stopped and then destroyed:



Try to create new Virtual machine with **C7** name, **2** CPU, **4GB** RAM and Static IPs **10.50.94.100:**
$ **ezmomi clone --template cos7box --hostname C7 --cpus 2 --mem 4 --ips 10.50.94.100**

Result:



Now we will try to provisioning to virtual machine. Create additional-provisioning-steps.sh script with the following contents. Script will install and add to startup httpd daemon and then add index.html file to the PUBLIC_HTML folder.
$ **cat additional-provisioning-steps.sh**

```
#!/usr/bin/env bash

IP="10.50.94.100"
pass="vagrant"
sleep 60
ssh-keyscan $IP >> ~/.ssh/known_hosts 2> /dev/null
sshpass -p "$pass" ssh root@"$IP" "yum -y install httpd"
sshpass -p "$pass" ssh root@"$IP" "systemctl start httpd && systemctl enable
httpd"
sshpass -p "$pass" ssh root@"$IP" "echo '<html><center><h1> Provisioning is
worked! </h1></center></html>' > /var/www/html/index.html"
```

Give exec access to this script:
$ **chmod +x additional-provisioning-steps.sh**

Create new virtual machine and provision this script after start:
$ **ezmomi clone --template cos7box --hostname C7 --cpus 2 --mem 4 --ips 10.50.94.100 --post-clone-cmd additional-provisioning-steps.sh**
Cloning cos7box to new host c7 with 4096MB RAM...
Running --post-clone-cmd ./additional-provisioning-steps.sh

End result must be as following: