
PVC Documentation

Release 0.1.5

Marin Atanasov Nikolov

May 11, 2015

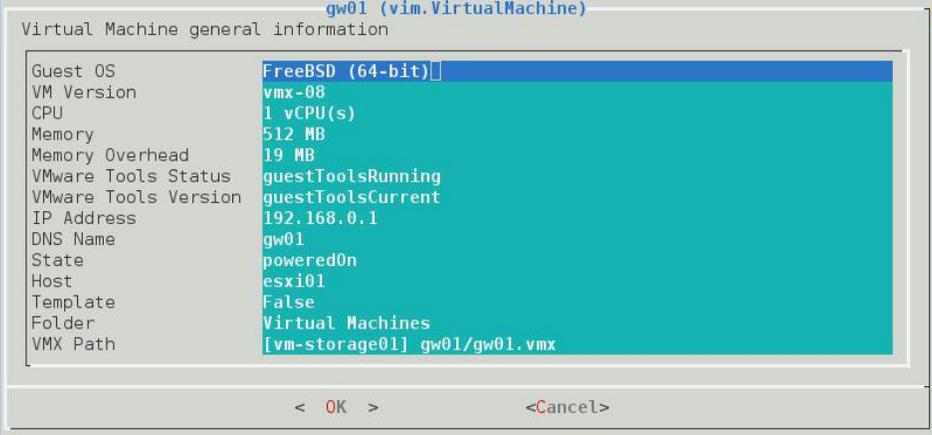
1	Status	3
2	Contributions	5
3	Bugs	7
4	Getting Started	9
5	Contents	11
5.1	Installation of PVC	11
5.2	Configuration of PVC	12
5.3	Example screenshots of PVC	13

PVC is an interactive text-mode VMware vSphere Client with a `dialog(1)` interface for GNU/Linux systems built on top of the `pyVmomi` VMware vSphere API Python bindings.

Using PVC allows you to quickly navigate in your VMware vSphere environment and perform common tasks against various VMware vSphere Managed Entities.

PVC is Open Source and licensed under the [BSD License](#).

```
vc01 - VMware vCenter Server 5.5.0 build-2442329 - Python vSphere Client version 0.1.0
```



```
gw01 (vim.VirtualMachine)
Virtual Machine general information
Guest OS           FreeBSD (64-bit)
VM Version         vmx-08
CPU               1 vCPU(s)
Memory            512 MB
Memory Overhead   19 MB
VMware Tools Status  guestToolsRunning
VMware Tools Version  guestToolsCurrent
IP Address        192.168.0.1
DNS Name          gw01
State             poweredOn
Host              esxi01
Template          False
Folder            Virtual Machines
VMX Path          [vm-storage01] gw01/gw01.vmx
< OK >          <Cancel>
```

Status

Experimental. PVC is in early development stage. Using PVC in a production environment is not (yet) recommended.

Contributions

PVC is hosted on [Github](#). Please contribute by reporting issues, suggesting features or by sending patches using pull requests.

Bugs

Probably. If you experience a bug issue, please report it to the PVC issue tracker on [Github](#).

Getting Started

Getting started with PVC is easy - simply go over the *Installation of PVC* and *Configuration of PVC* pages, which provide all the details about how to install and configure PVC.

Make sure to also check the *Example screenshots of PVC* page, which contains example screenshots of using PVC in a VMware vSphere environment.

5.1 Installation of PVC

This document walks you through the installation of PVC.

The easiest way to install PVC is by using `pip`, which would automatically install any dependencies for you or you could use the latest development version of PVC from the [Github](#) repository.

5.1.1 Requirements

The following list provides information about the PVC dependencies.

- [Python 2.7.x, 3.2.x or later](#)
- [humanize](#)
- [pythondialog](#)
- [pyVmomi](#)
- [requests](#)
- [vconnector](#)

Some of the PVC features require additional packages to be present in order to take advantage of these features. The list below provides information about any optional dependencies of PVC.

Note, that these dependencies are not required and are only needed if you intend to use the features provided by them.

- [gnuplot](#) - Used for plotting performance graphs
- [VMware Player](#) - Used for establishing a remote console session
- A VNC client - Used for establishing a remote console VNC session

5.1.2 Installation with pip

In order to install PVC using `pip`, simply execute this command:

```
$ pip install pvc
```

If you would like to install PVC in a `virtualenv`, then follow these steps instead:

```
$ virtualenv pvc-venv
$ source pvc-venv/bin/activate
$ pip install pvc
```

5.1.3 Installation from source

The `master` branch of PVC is where main development happens.

In order to install the latest version of PVC follow these simple steps:

```
$ git clone https://github.com/dnaeon/pvc.git
$ cd pvc
$ sudo python setup.py install
```

If you would like to install PVC in a `virtualenv` follow these steps instead:

```
$ virtualenv pvc-venv
$ source pvc-venv/bin/activate
$ git clone https://github.com/dnaeon/pvc.git
$ cd pvc
$ python setup.py install
```

This would take care of installing all dependencies for you as well.

5.2 Configuration of PVC

By default PVC does not require any special configuration to work.

5.2.1 Gnuplot Configuration Options

If you are using `gnuplot` for plotting performance graphs with PVC and you want to customize the `gnuplot` terminal being used you could set the `GNUPLOT_TERM` environment variable to your desired terminal.

If `GNUPLOT_TERM` is not set then PVC will use a `dumb` terminal when plotting a performance graph.

5.2.2 VMRC Console Support

Launching a remote console to a Virtual Machine requires that you have `VMRC` installed on your system.

Currently `VMRC` support is provided by VMware for Windows(R) and Mac OS X systems and support for GNU/Linux is underway.

`VMRC` support for GNU/Linux is planned, but not yet available, so in order to launch a console to a Virtual Machine from a GNU/Linux system you need to use `VMware Player` for now.

Check [KB 2091284](#) for more details on the `VMRC` support.

5.2.3 VNC Console Support

PVC supports launching of VNC console to a Virtual Machine, but you need to make sure that certain ports on the ESXi hosts are opened, so that a successful VNC connection can be established.

PVC uses ports 5901-5999 for establishing a VNC connection to a Virtual Machine.

Refer to [How to Create Custom Firewall Rules in ESXi 5.0](#) article for more information on how to manage the firewall rules on your VMware ESXi hosts and open the required ports for VNC communication.

5.3 Example screenshots of PVC

In this page you will find some example screenshots of using PVC in a VMware vSphere environment.

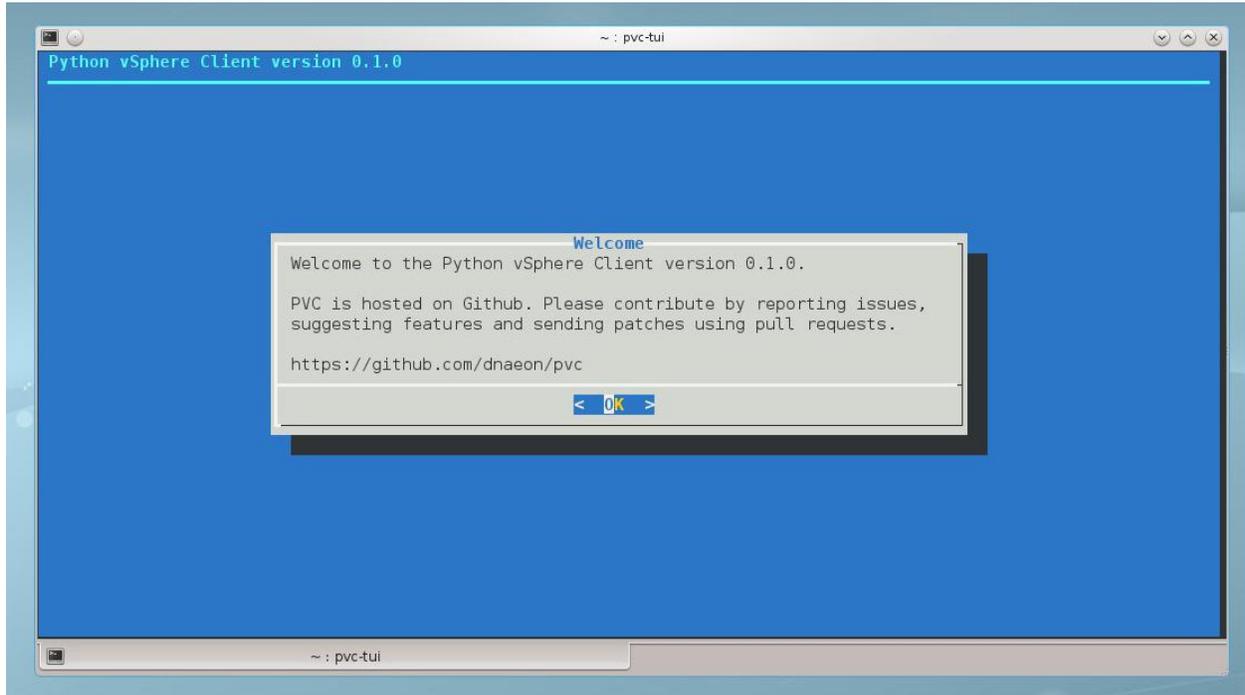


Fig. 5.1: The PVC Welcome screen

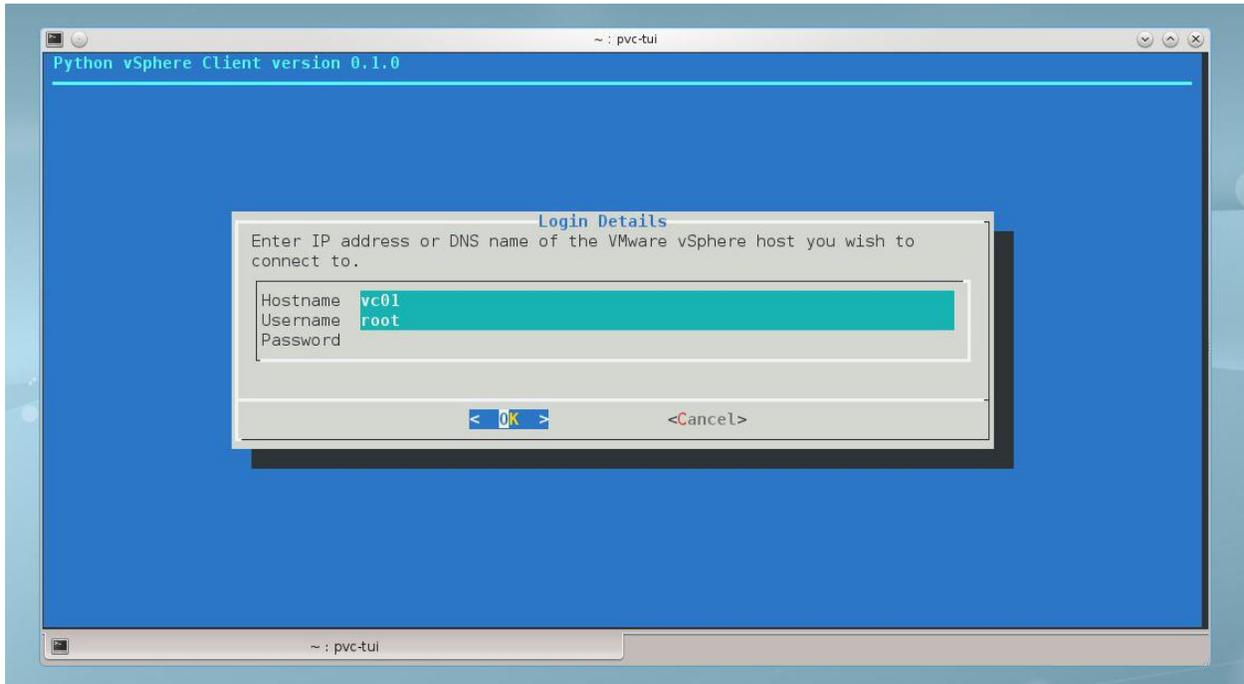


Fig. 5.2: Login window for establishing a connection to a VMware vSphere host

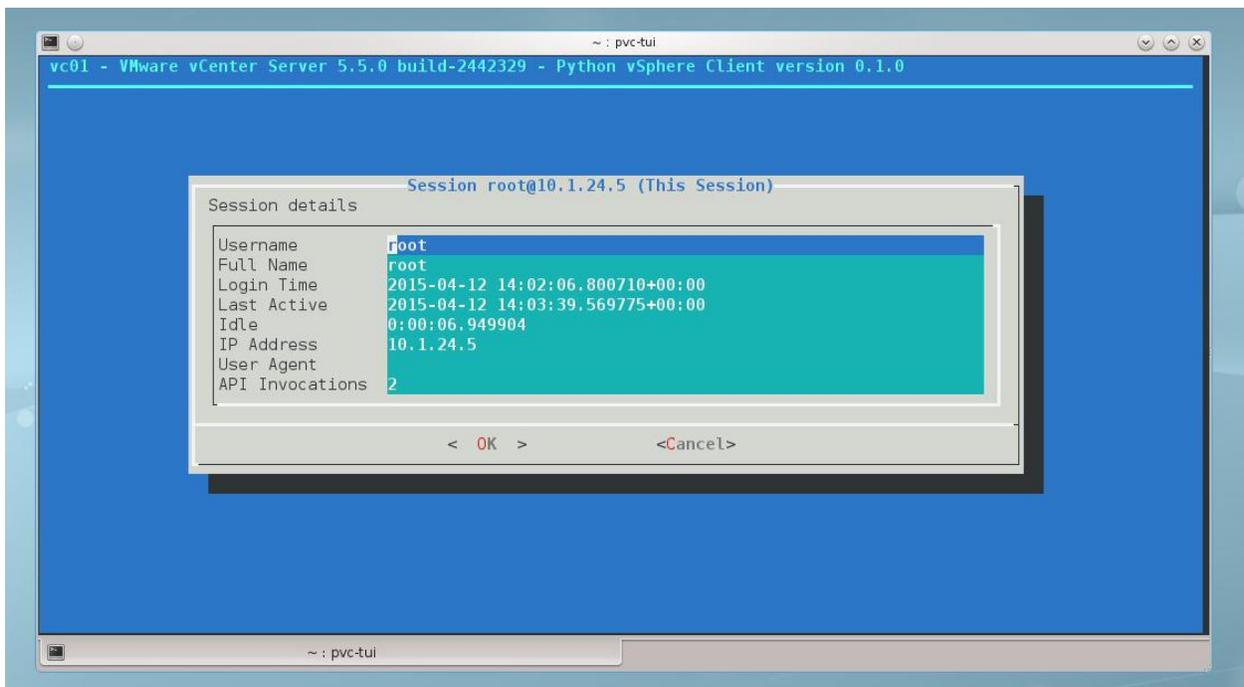


Fig. 5.3: PVC displaying details about an established vSphere session

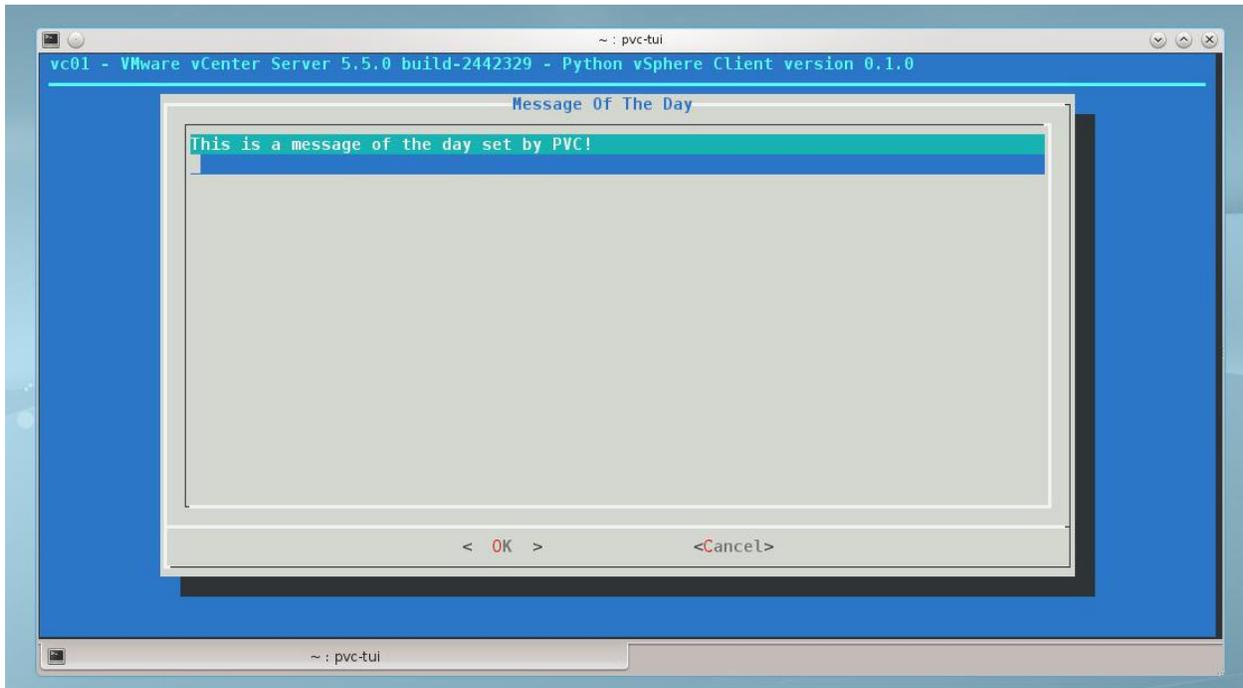


Fig. 5.4: PVC displaying a message of the day, set by PVC!

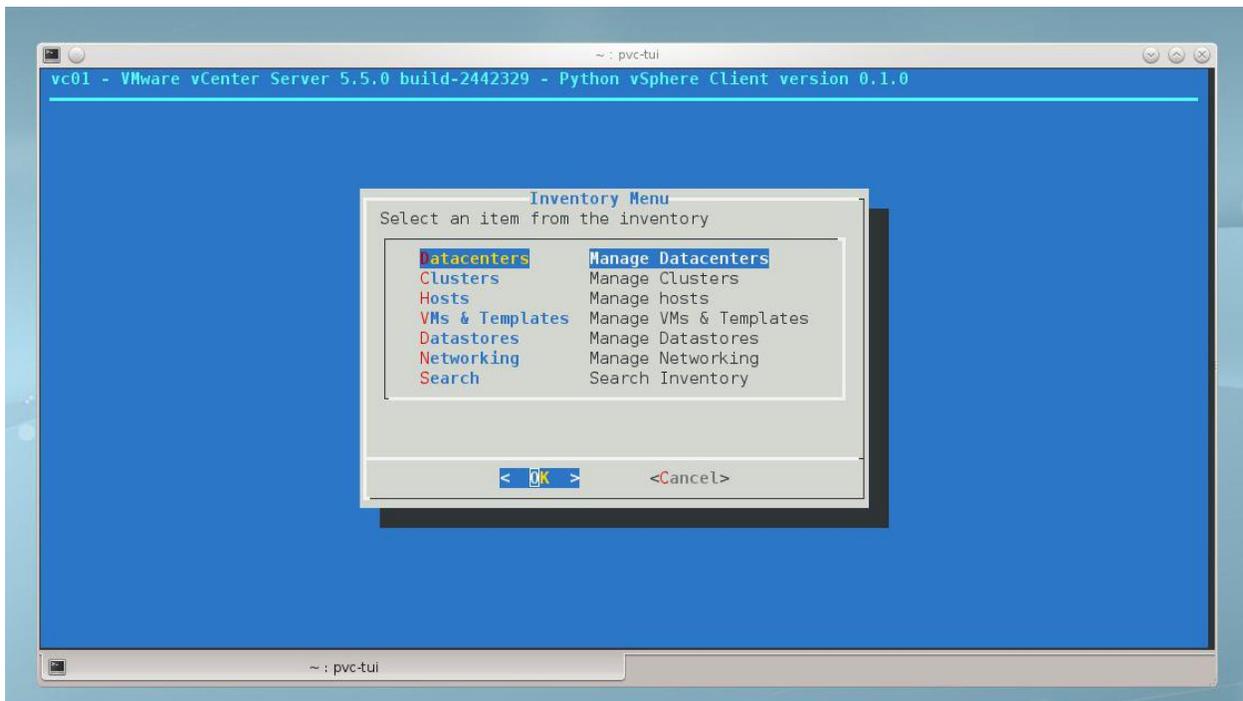


Fig. 5.5: The inventory menu of PVC and the available inventory items

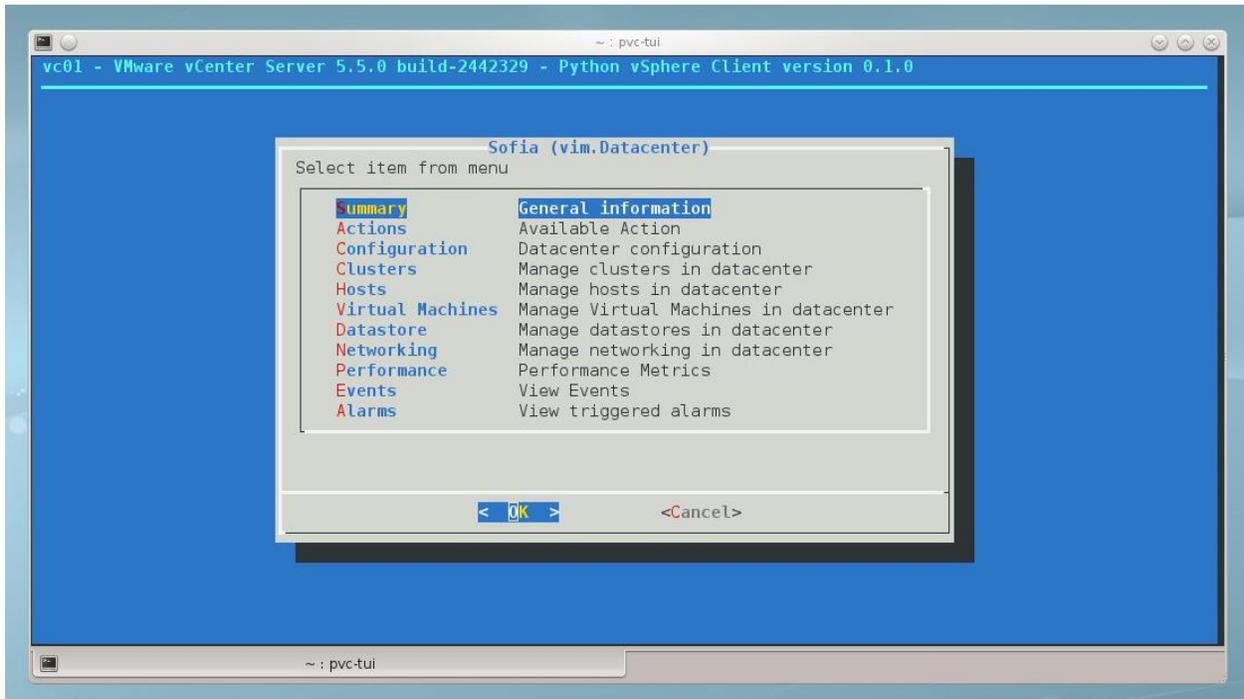


Fig. 5.6: The menu of available actions when browsing in a vSphere Datacenter

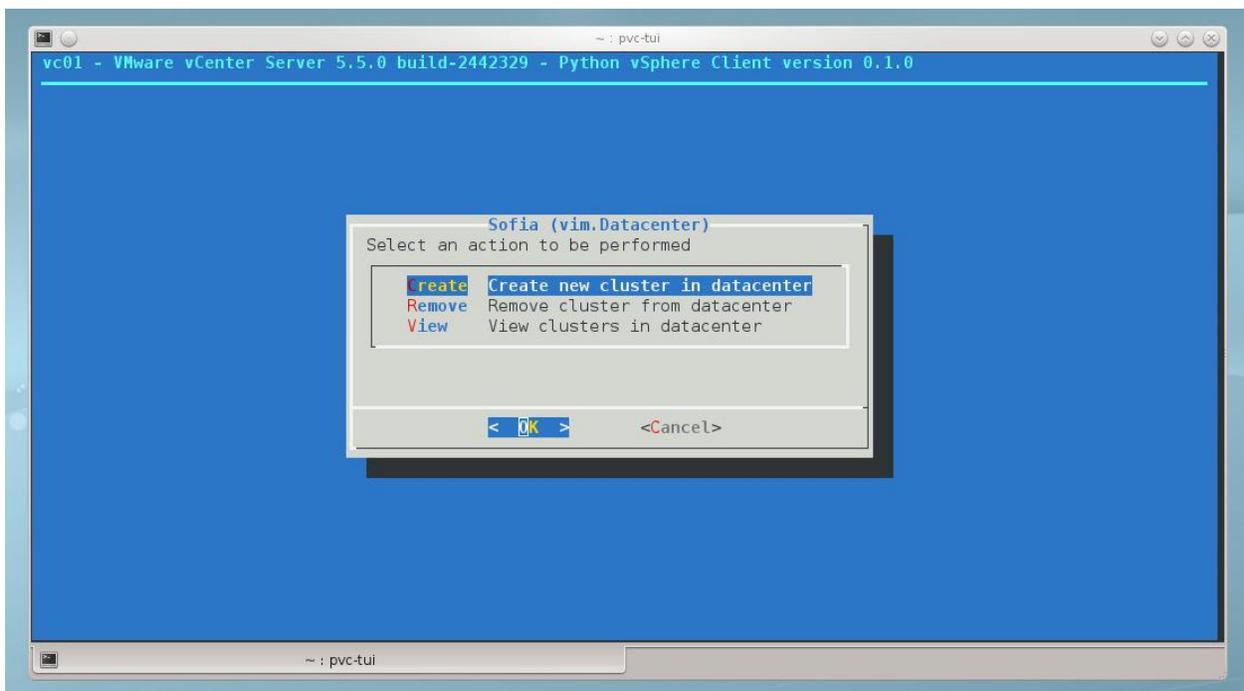


Fig. 5.7: PVC menu allowing you to create, remove and view clusters within a VMware vSphere Datacenter

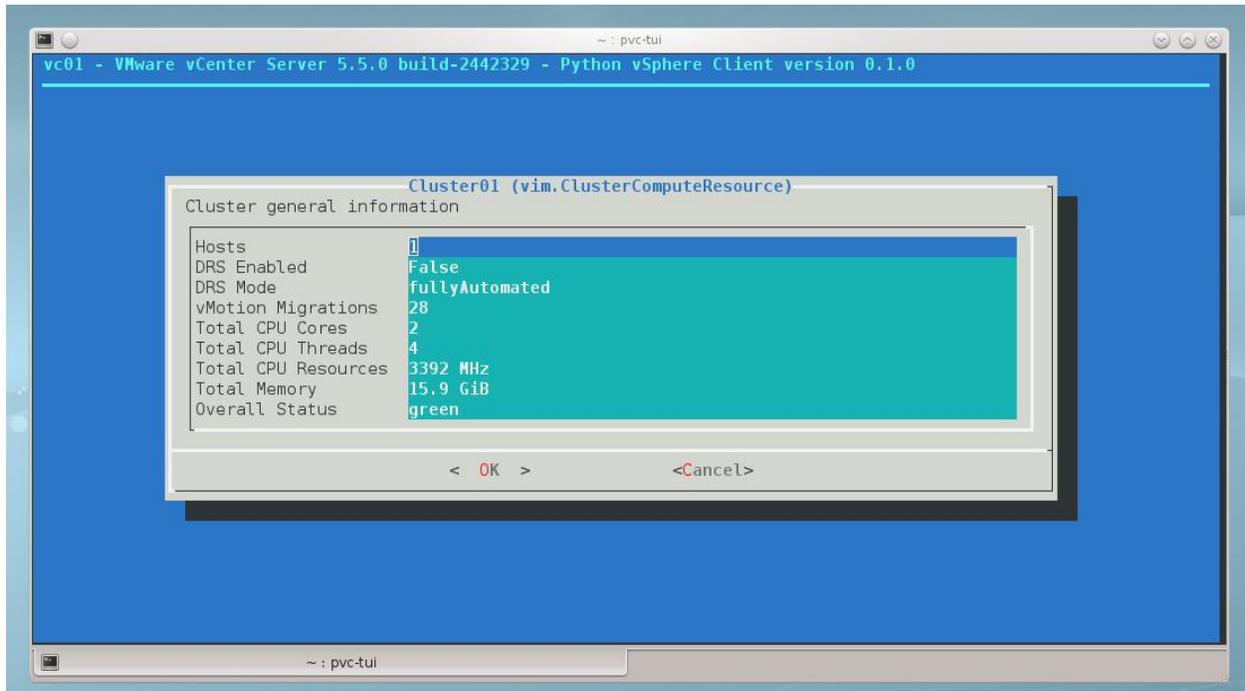


Fig. 5.8: PVC displaying the summary information about a VMware vSphere Cluster

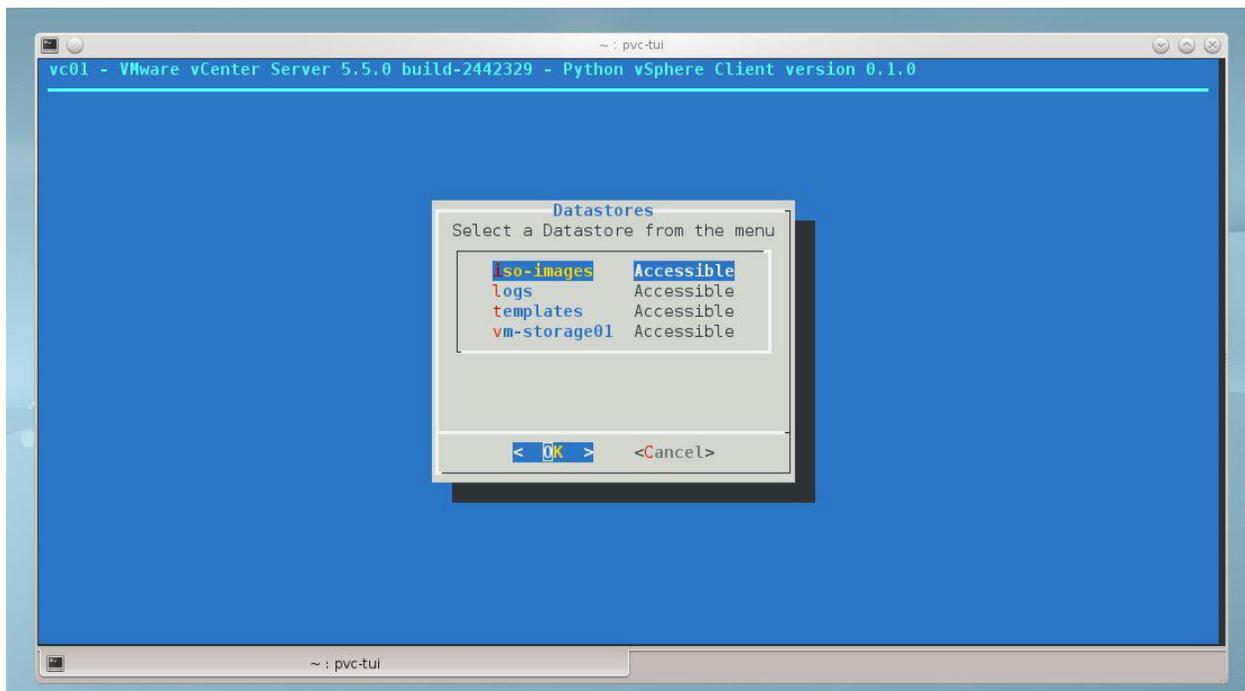


Fig. 5.9: PVC displaying a menu of discovered Datastores and their connection state

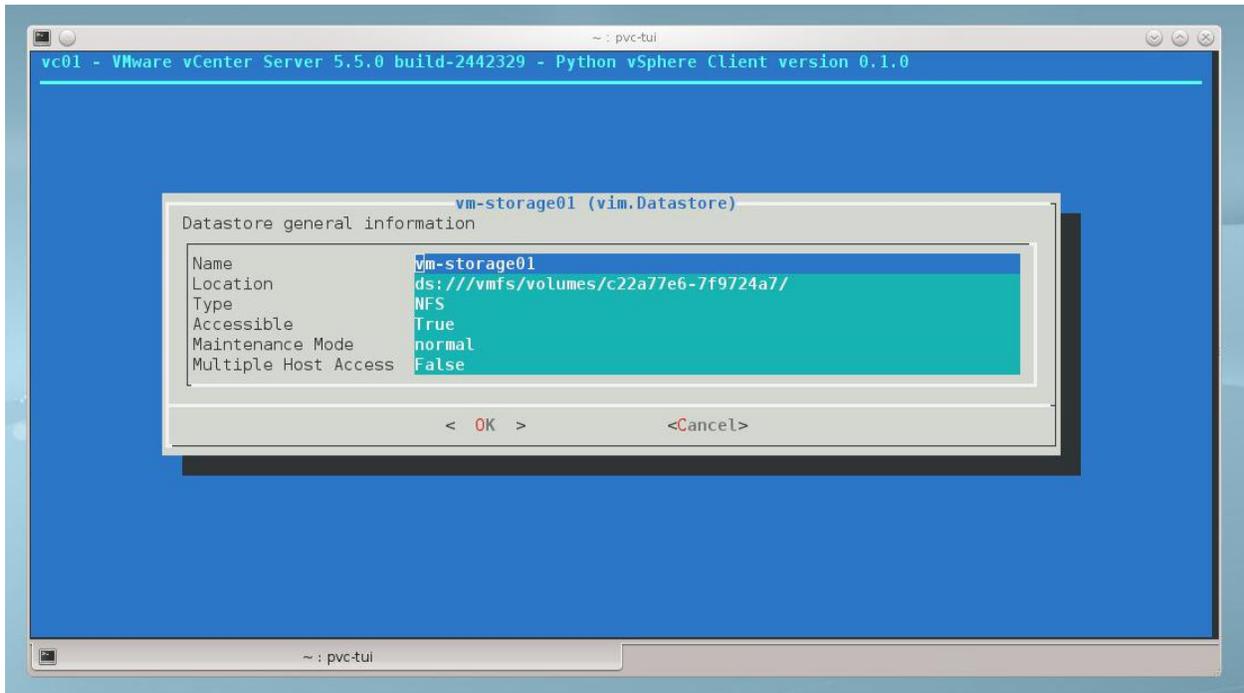


Fig. 5.10: PVC displaying summary information about a datastore

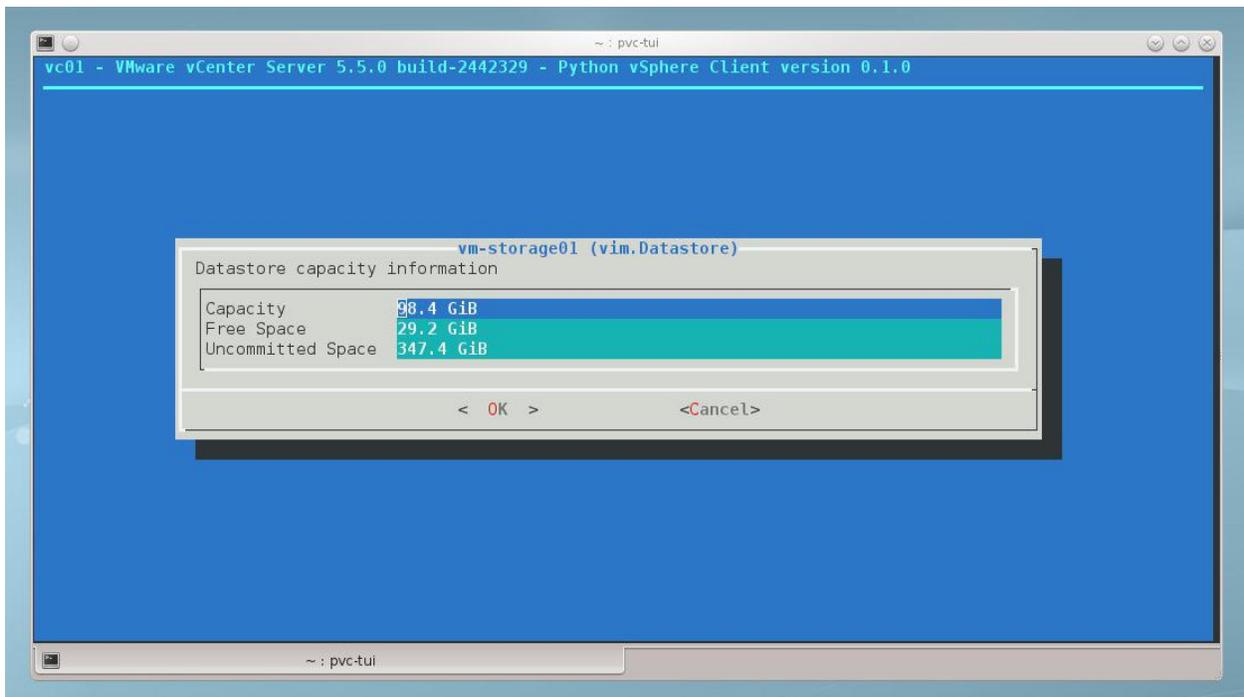


Fig. 5.11: PVC displaying capacity information about a datastore

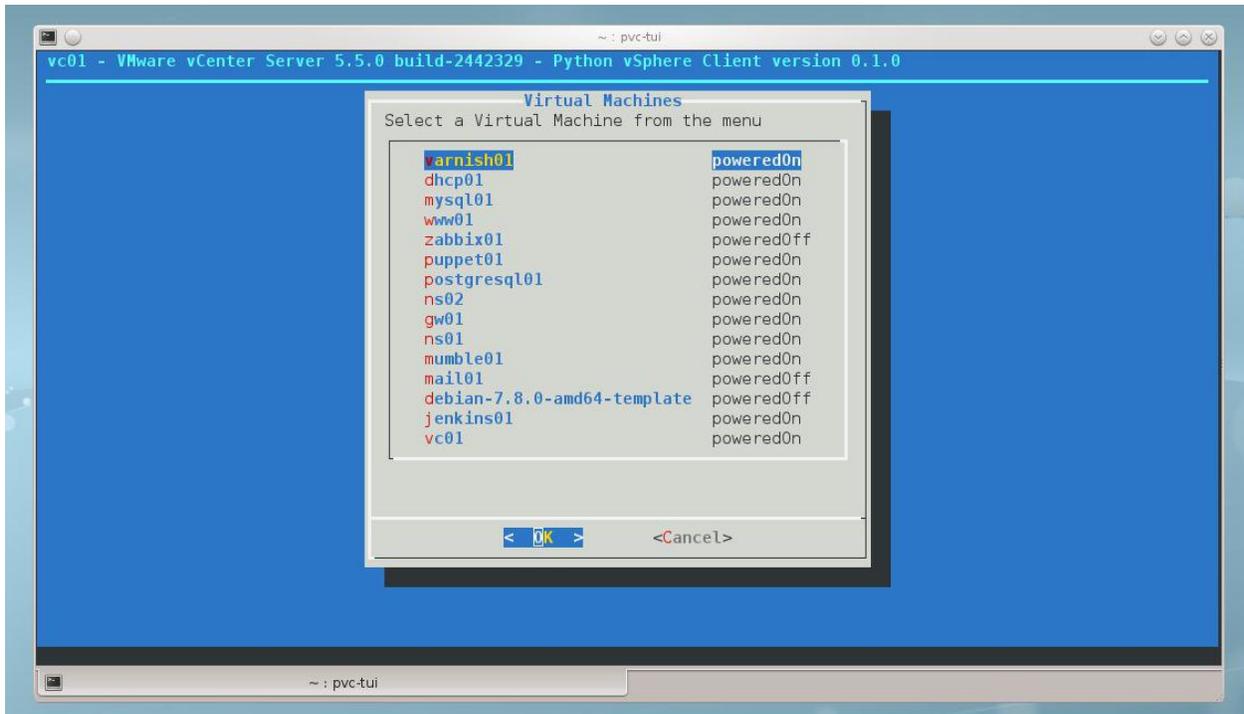


Fig. 5.12: PVC displaying a menu of Virtual Machines and their power state

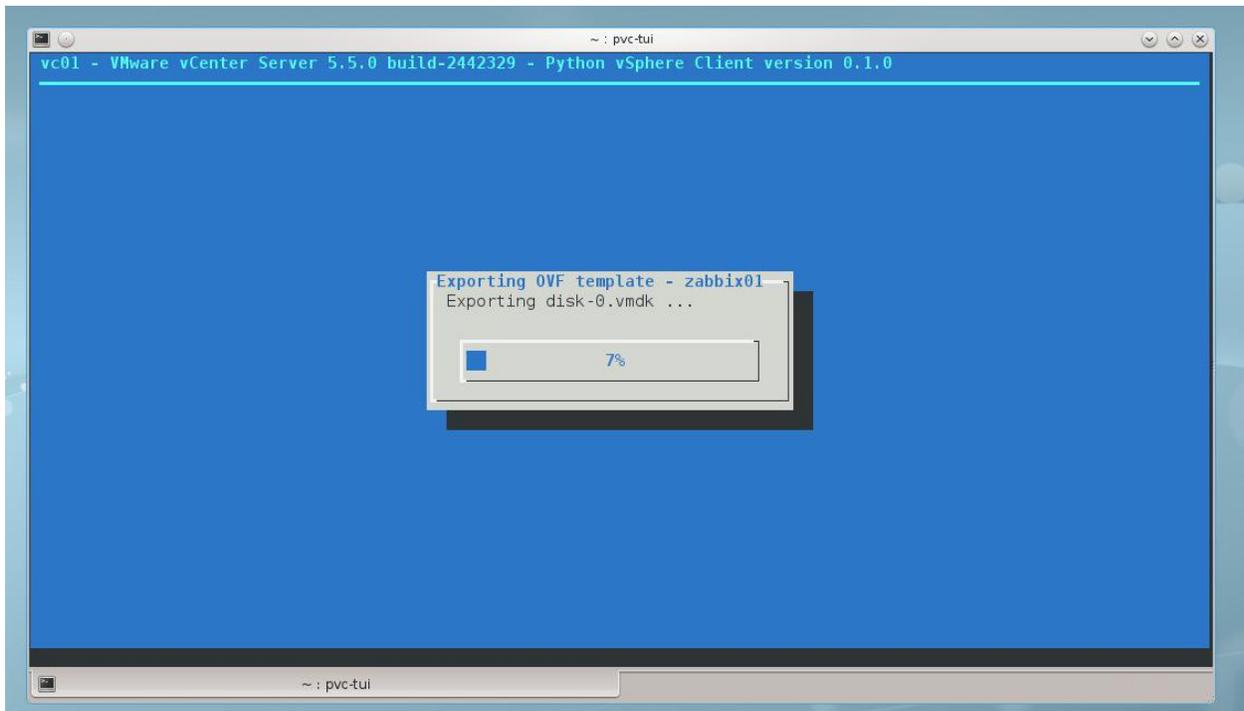


Fig. 5.13: PVC displaying the progress of exporting a Virtual Machine as OVF template

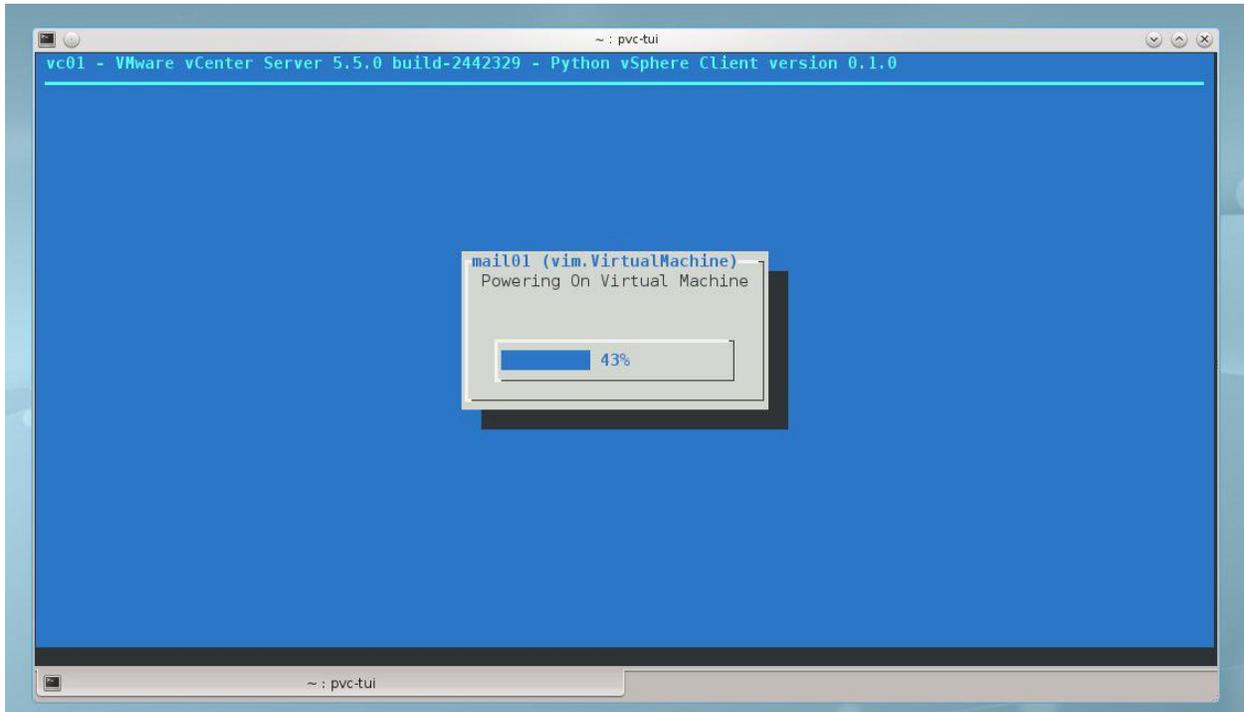


Fig. 5.14: PVC displaying the progress of powering on a Virtual Machine

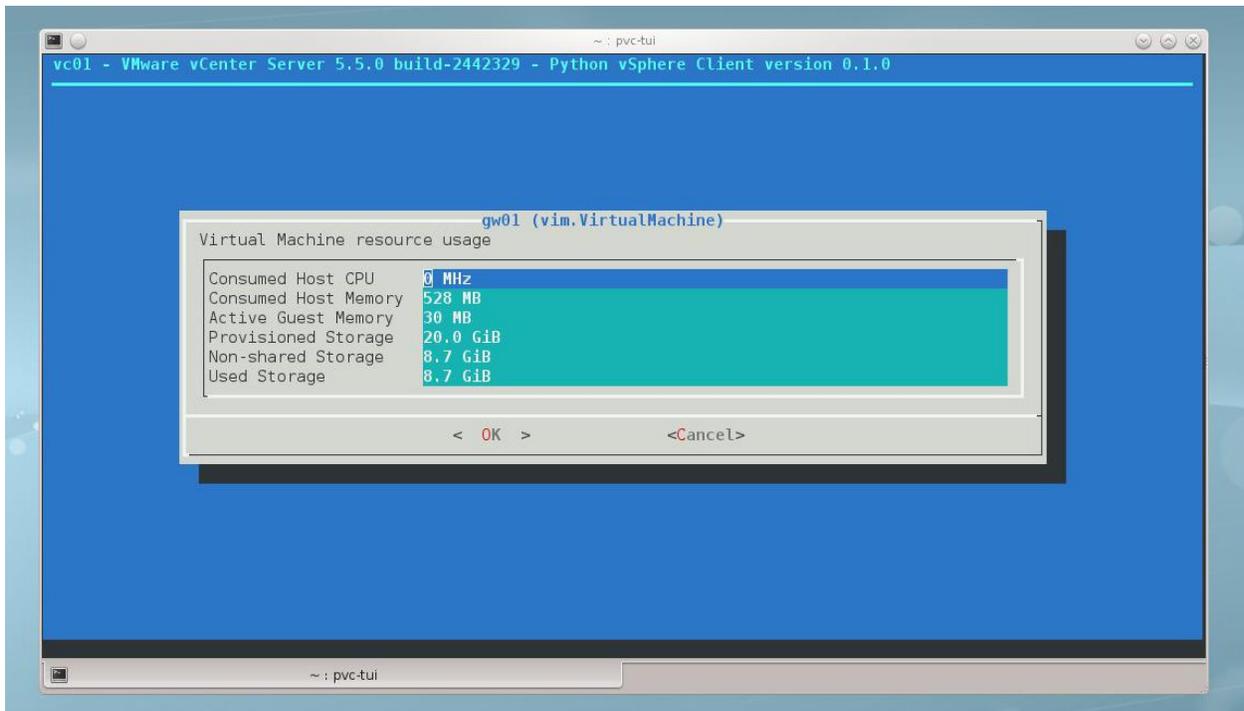


Fig. 5.15: PVC displaying the resource usage of a Virtual Machine

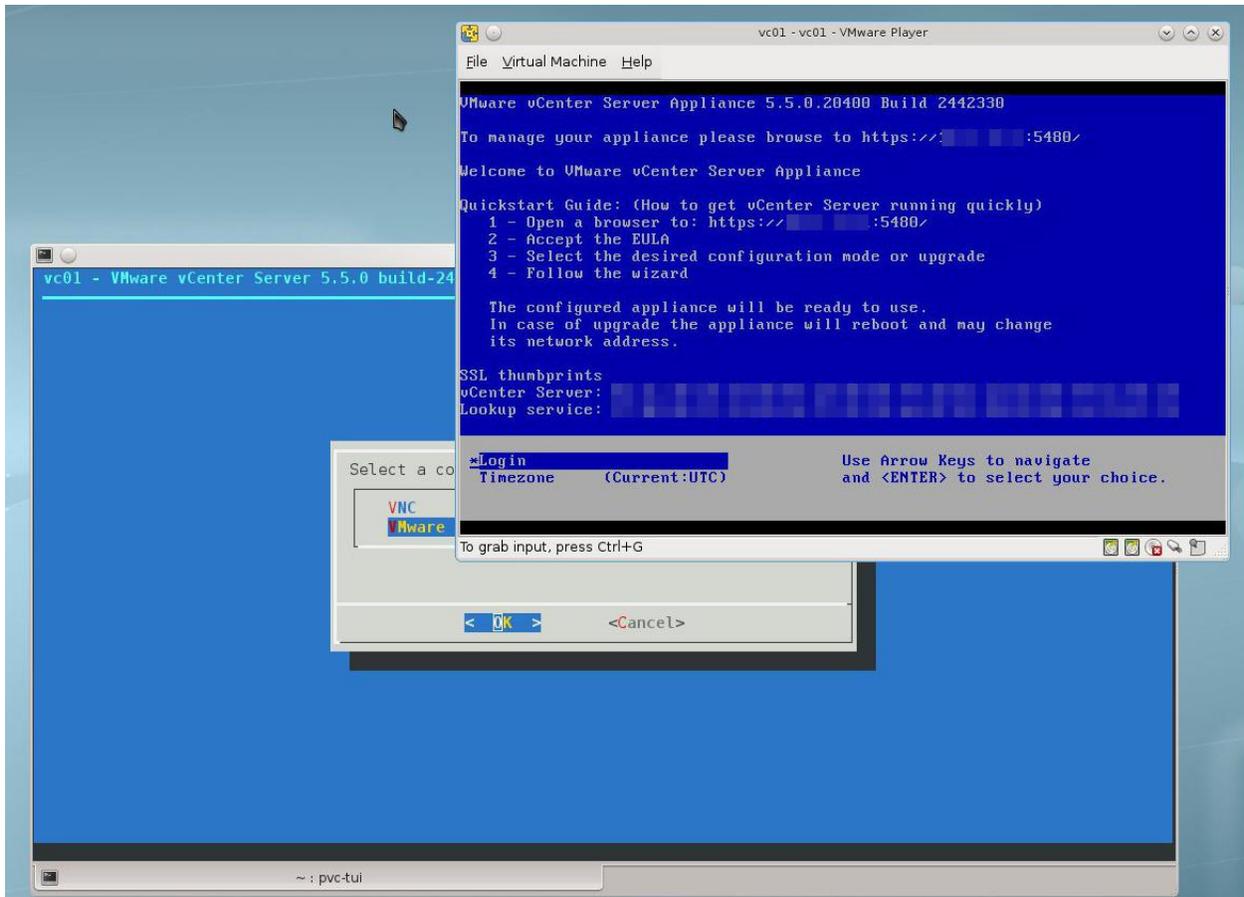


Fig. 5.16: A screenshot showing PVC successfully launching a VMware Player console to a Virtual Machine

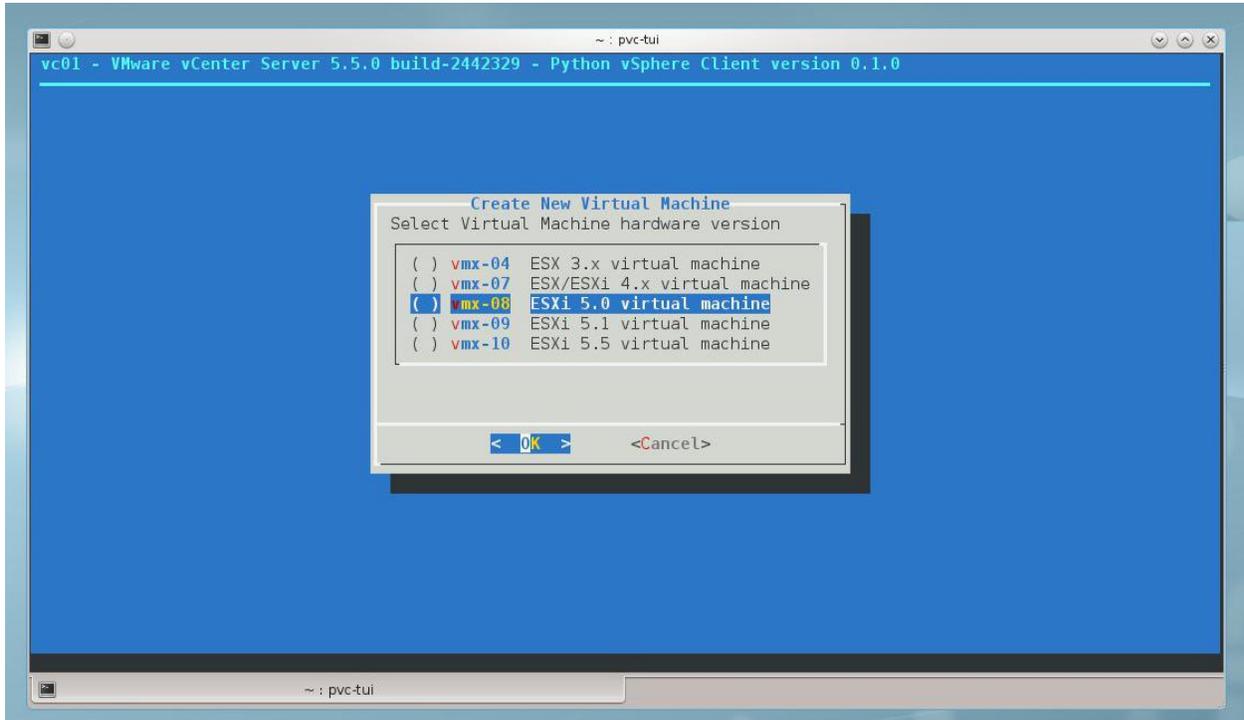


Fig. 5.17: PVC displaying a menu of available hardware versions to choose from during creation of a new Virtual Machine.

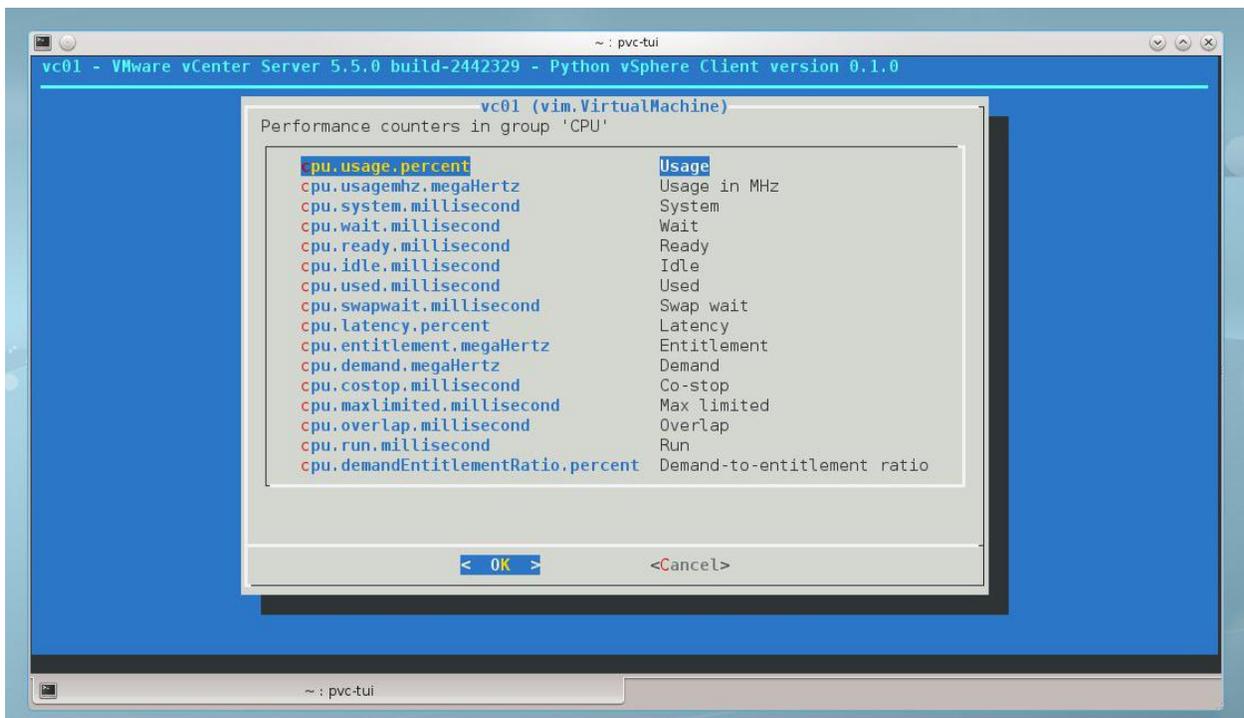


Fig. 5.18: PVC displaying a menu of available real-time performance counters on a Virtual Machine