# PVC Documentation

***Release 0.1.0***

**Marin Atanasov Nikolov**

April 17, 2015

Contents


**1 Status** **3**

**2 Contributions** **5**

**3 Bugs** **7**

**4 Getting Started** **9**

**5 Contents** **11**

5.1 Installation of PVC . . . . . 11

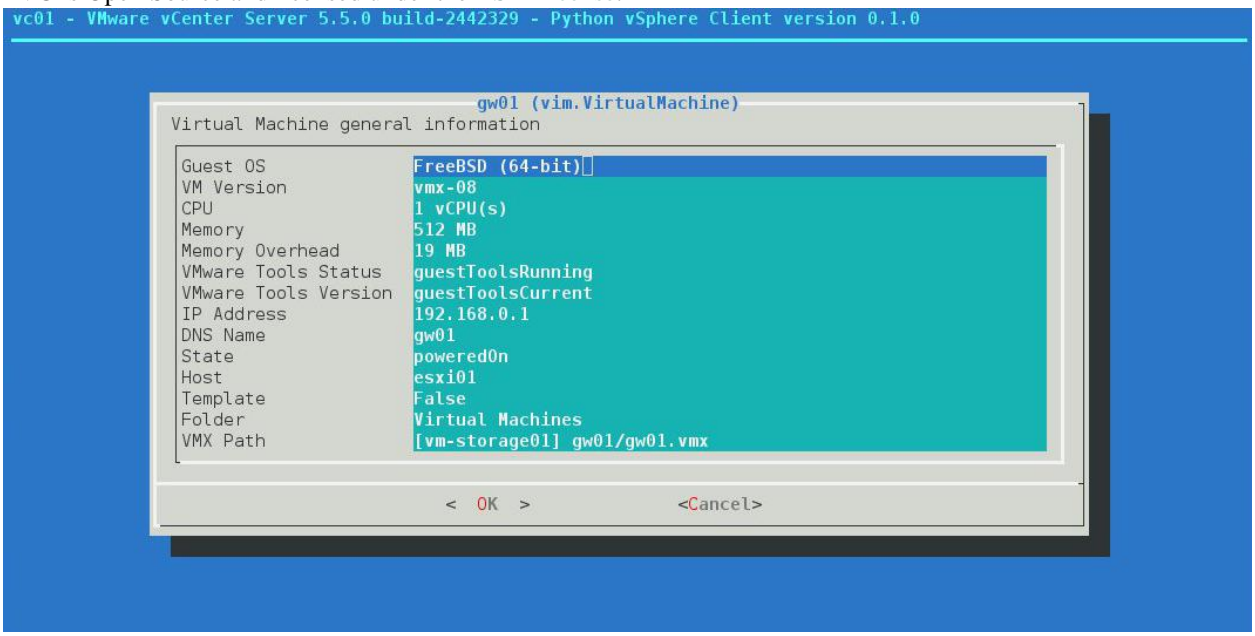
5.2 Configuration of PVC . . . . . 12

5.3 Example screenshots of PVC . . . . . 12

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.



CHAPTER 1

# Status

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

CHAPTER 2

# Contributions

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

CHAPTER 3

# Bugs

Probably. If you experience a bug issue, please report it to the PVC issue tracker on Github.

CHAPTER 4

# 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.

CHAPTER 5

# Contents

## 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 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.

If you are using gnuplot for plotting perfomance 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.

## 5.3 Example screenshots of PVC

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

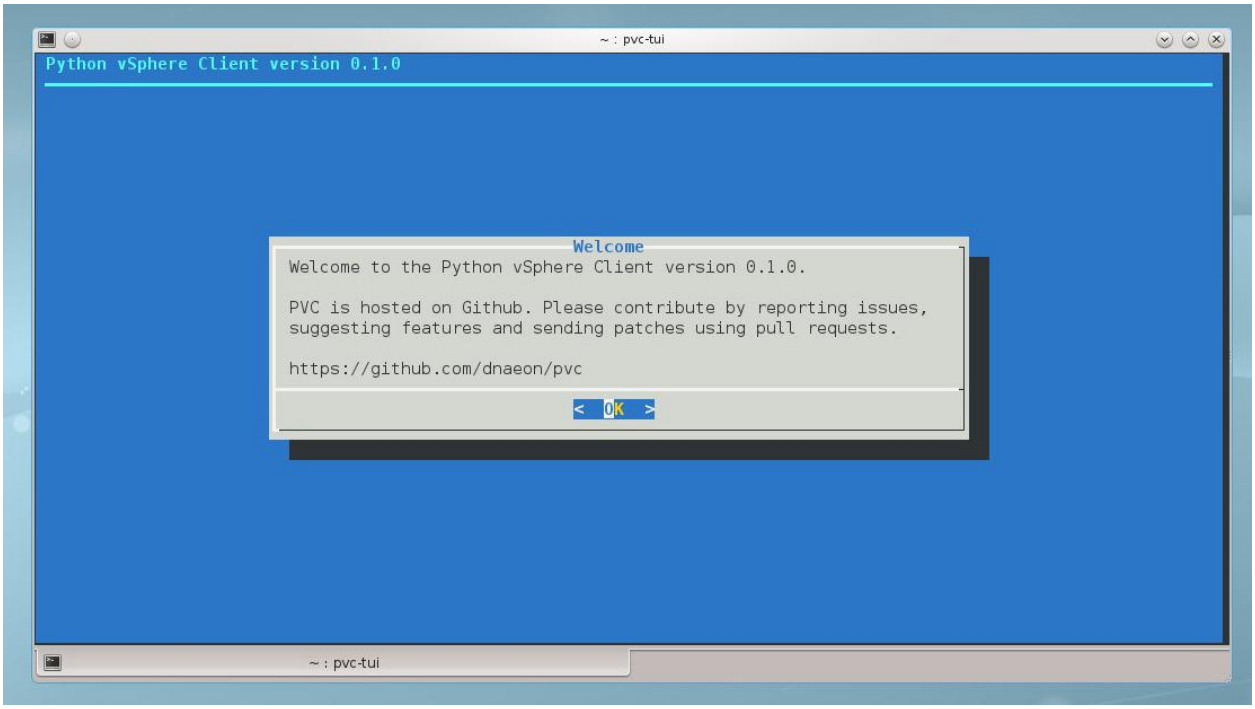



Figure 5.1: The PVC Welcome screen

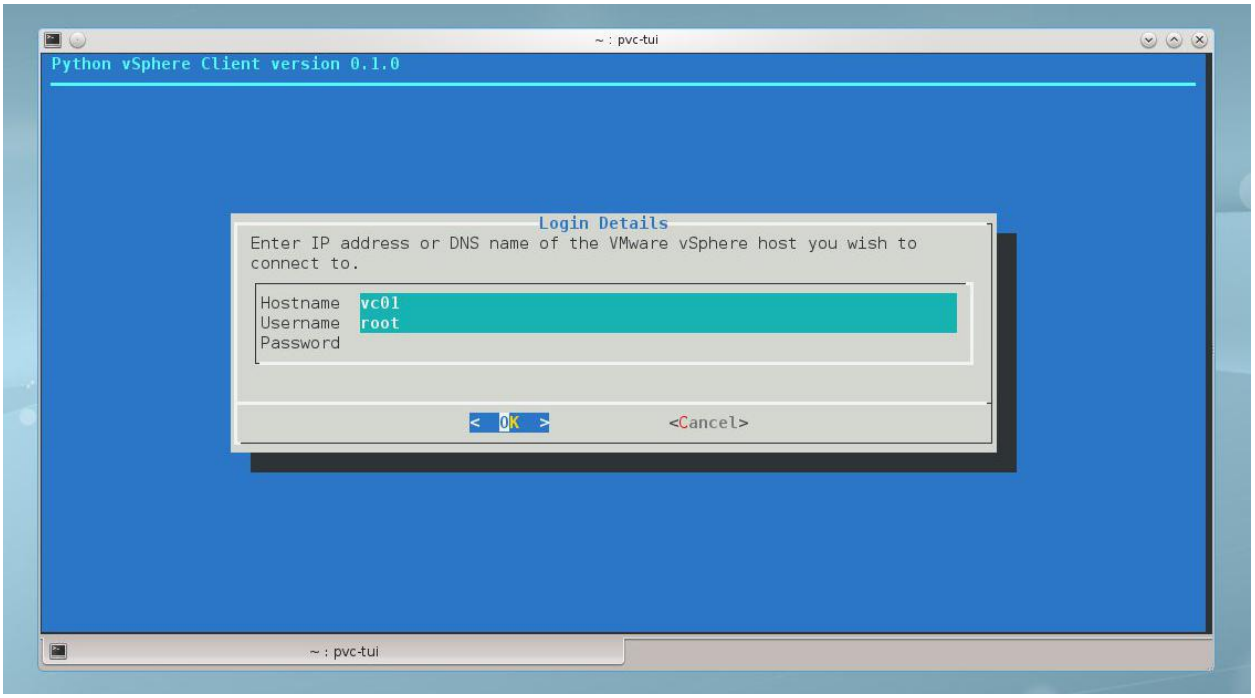



Figure 5.2: Login window for establishing a connection to a VMware vSphere host

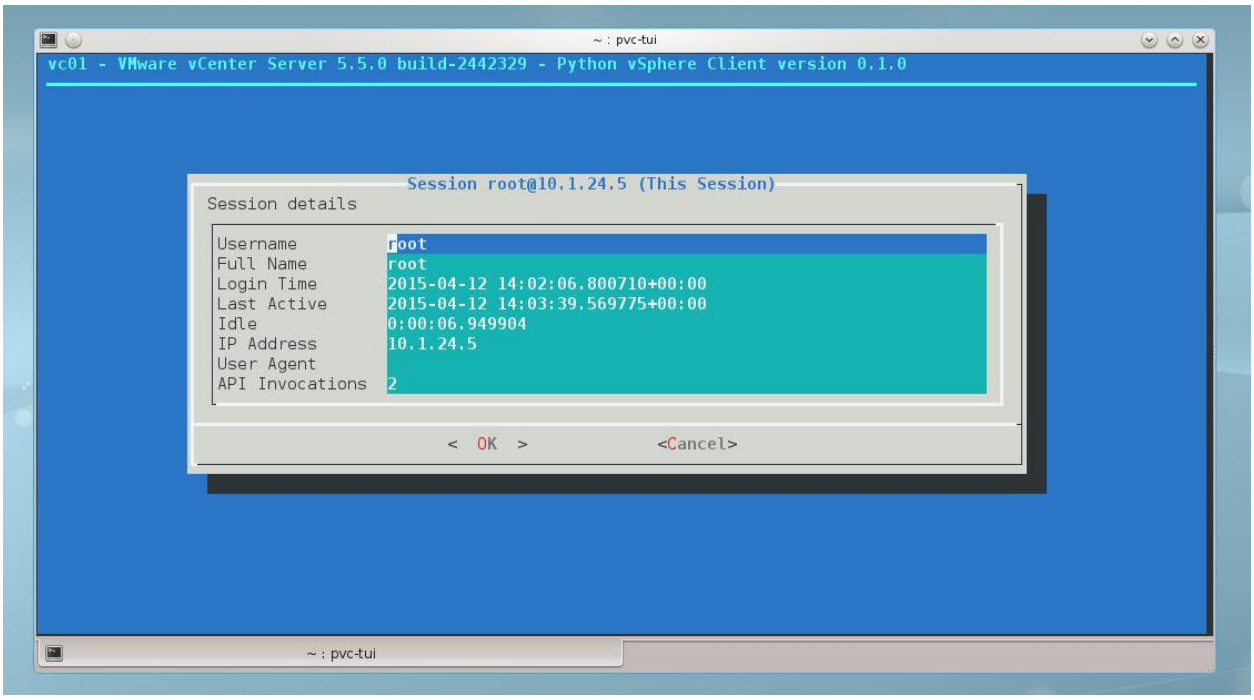



Figure 5.3: PVC displaying details about an established vSphere session

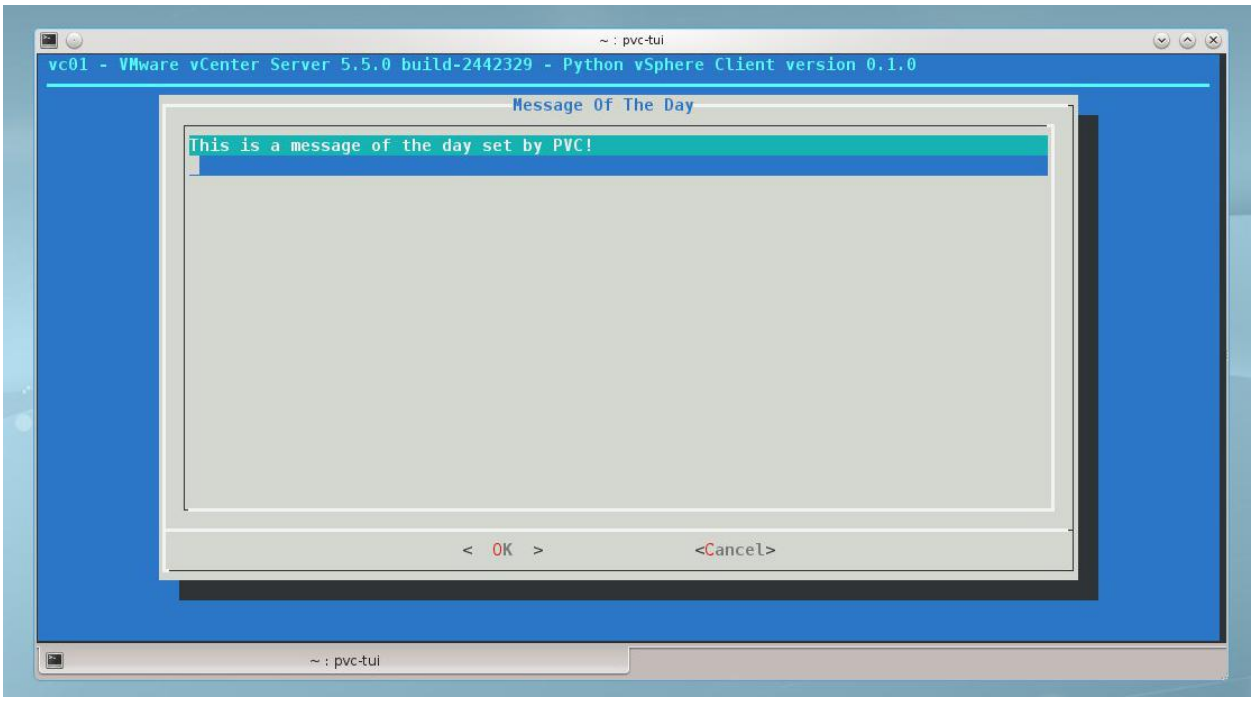



Figure 5.4: PVC displaying a message of the day, set by PVC!

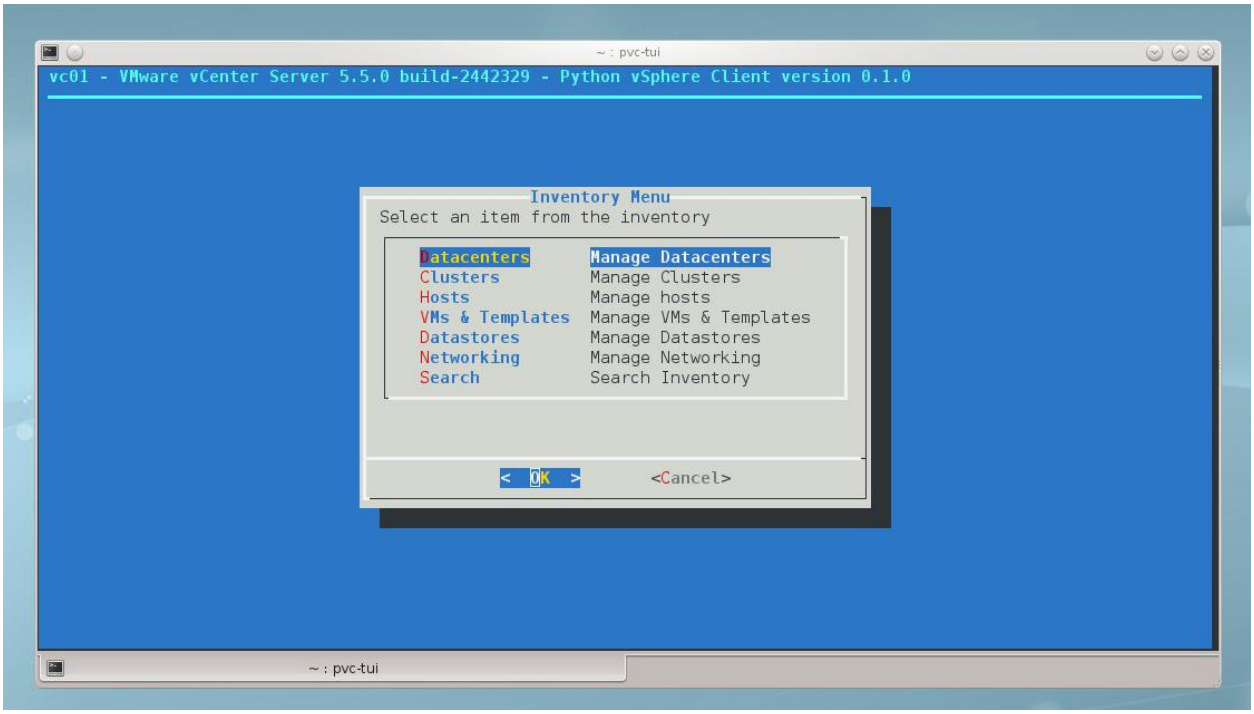



Figure 5.5: The inventory menu of PVC and the available inventory items

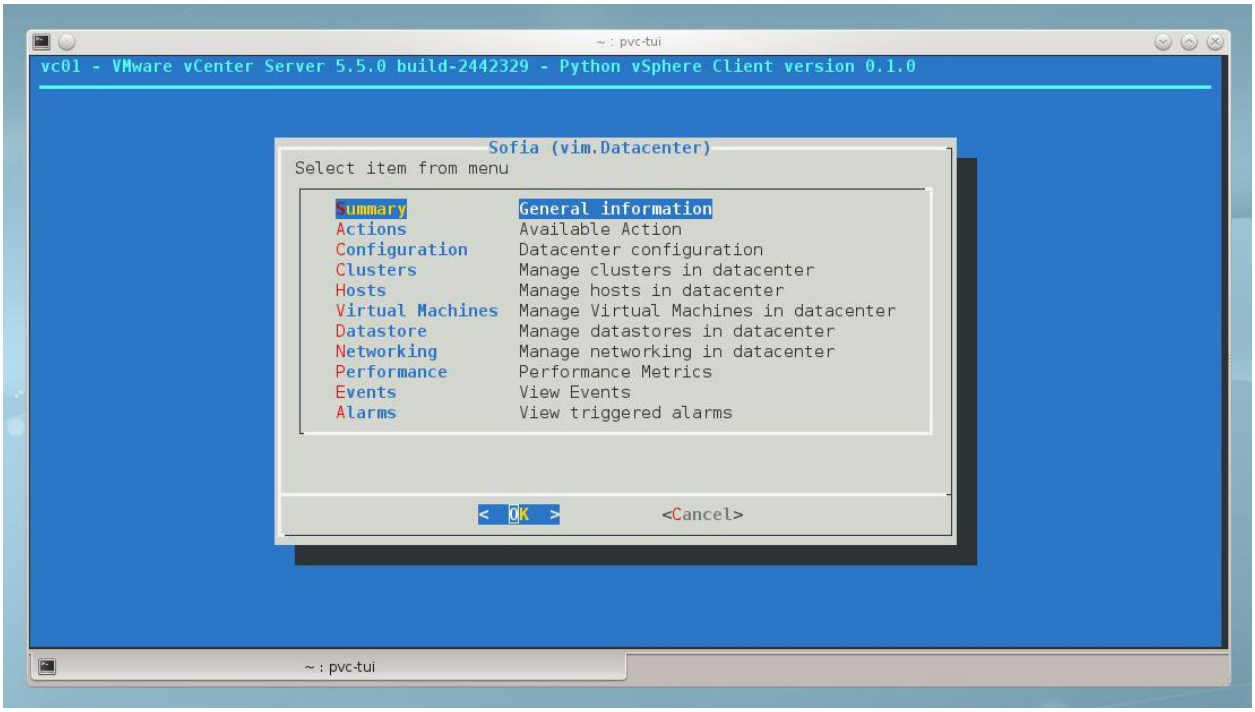



Figure 5.6: The menu of available actions when browsing in a vSphere Datacenter

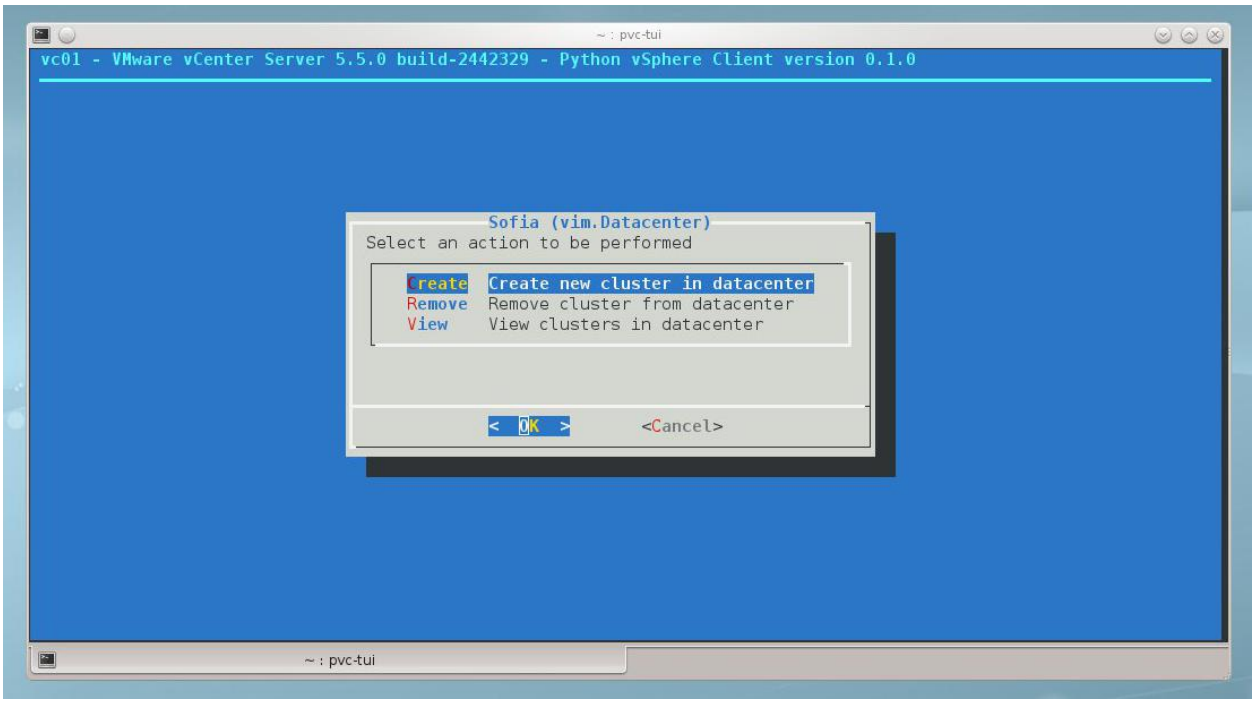



Figure 5.7: PVC menu allowing you to create, remove and view clusters within a VMware vSphere Datacenter

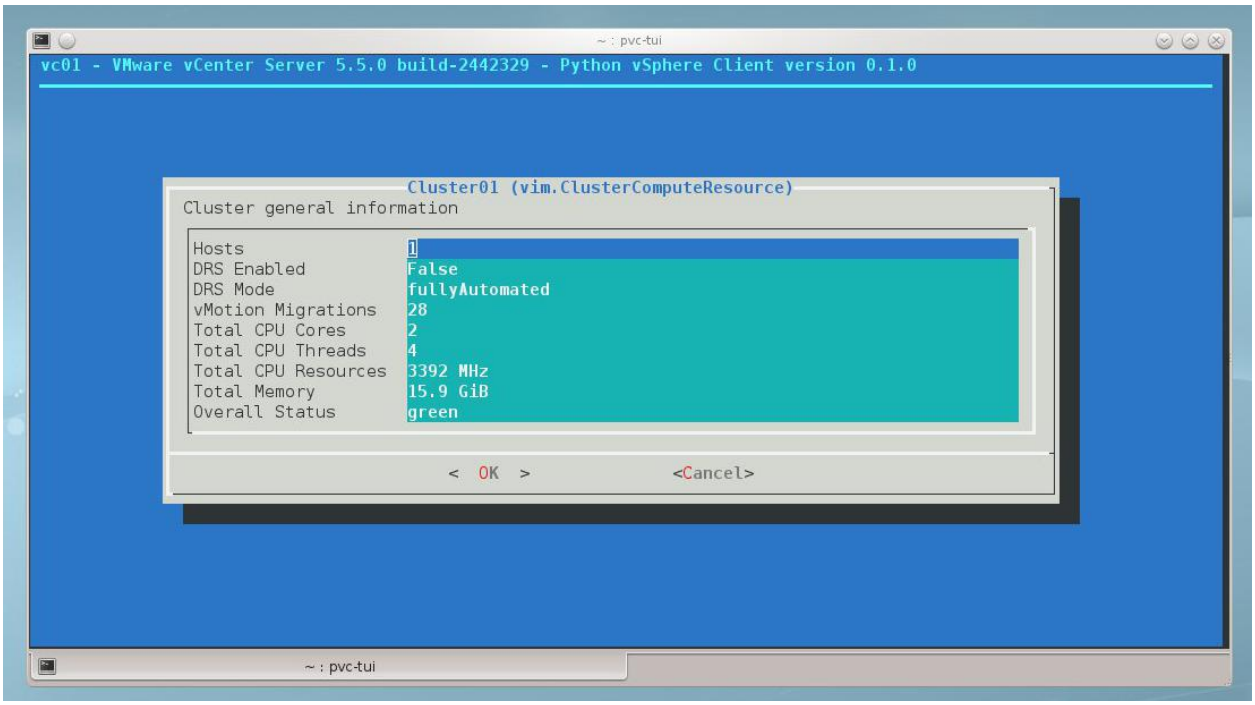



Figure 5.8: PVC displaying the summary information about a VMware vSphere Cluster

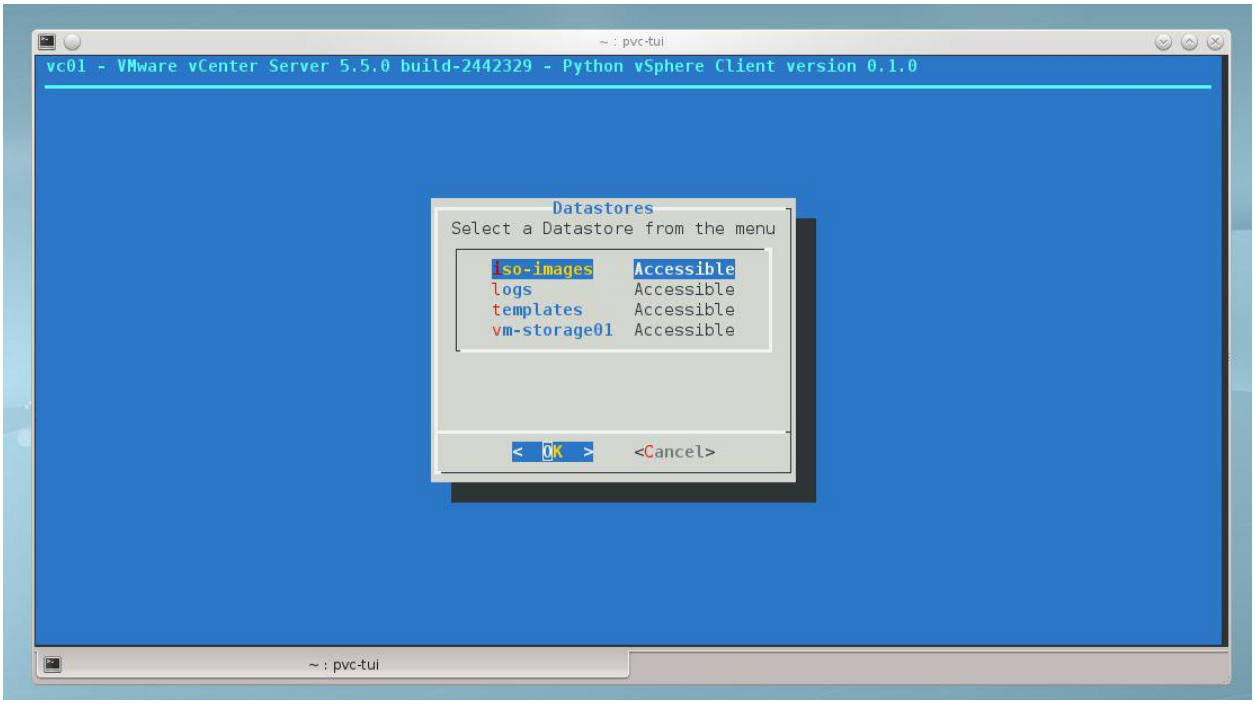



Figure 5.9: PVC displaying a menu of discovered Datastores and their connection state

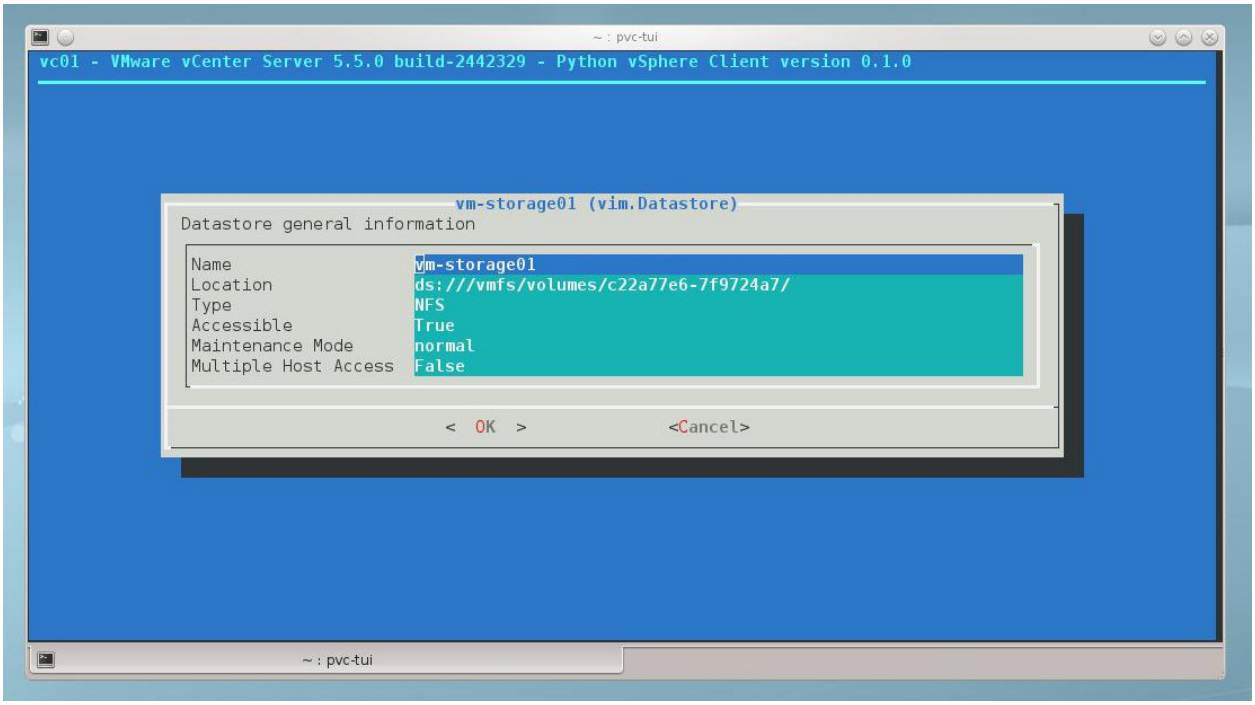



Figure 5.10: PVC displaying summary information about a datastore

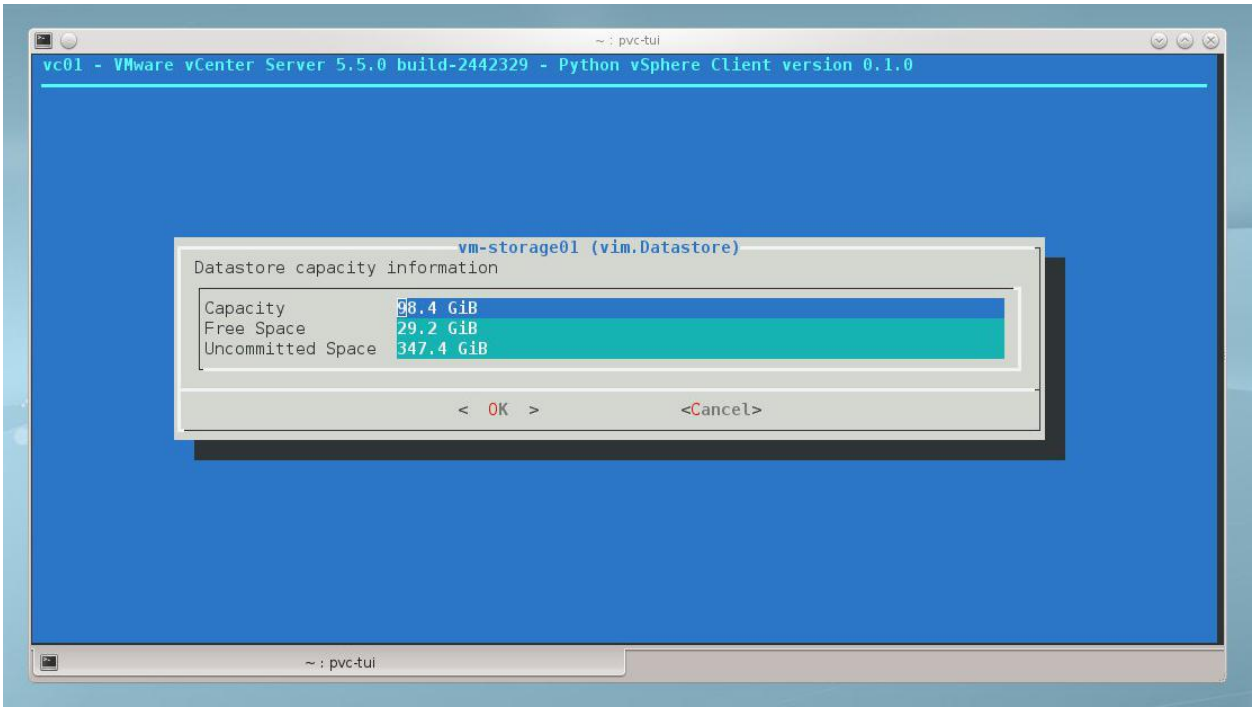



Figure 5.11: PVC displaying capacity information about a datastore

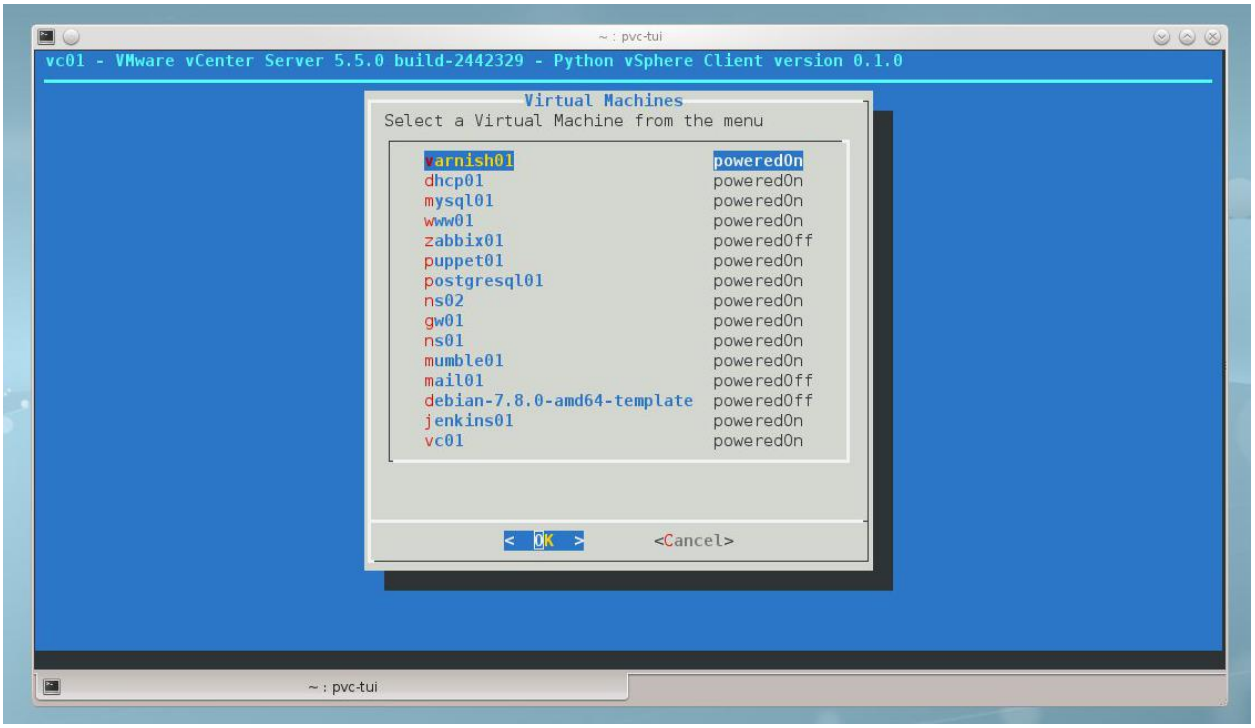



Figure 5.12: PVC displaying a menu of Virtual Machines and their power state

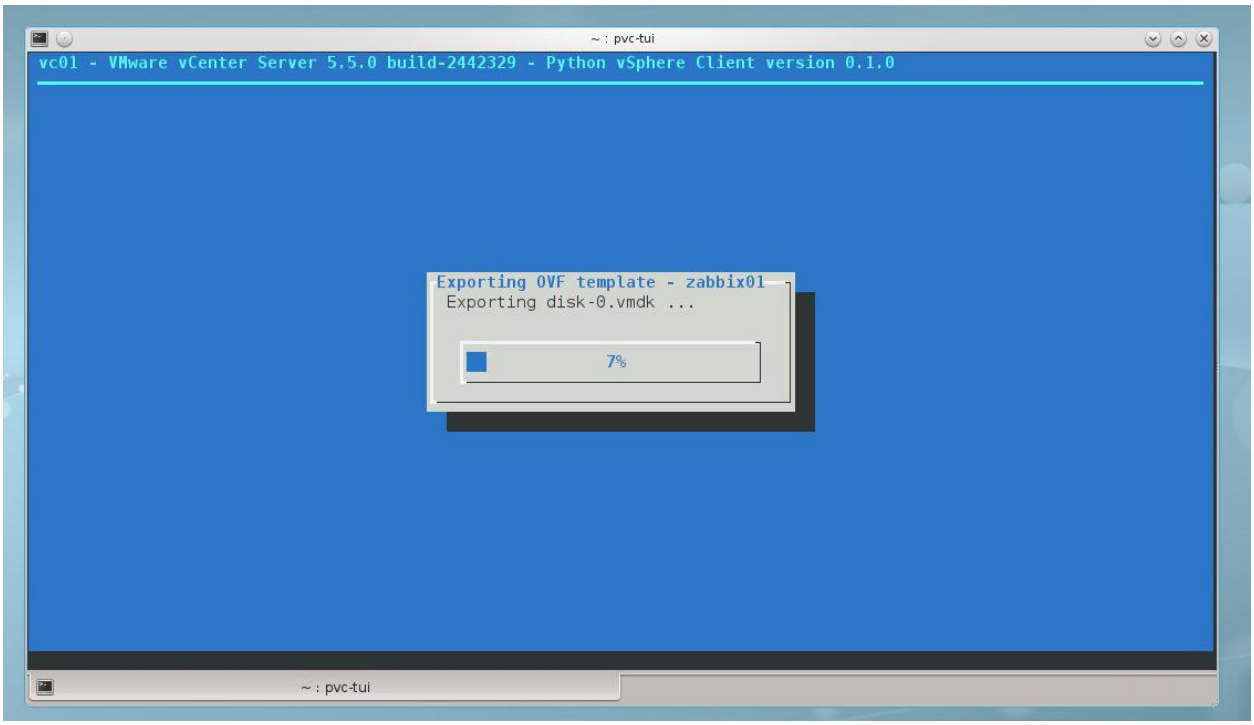



Figure 5.13: PVC displaying the progress of exporting a Virtual Machine as OVF template

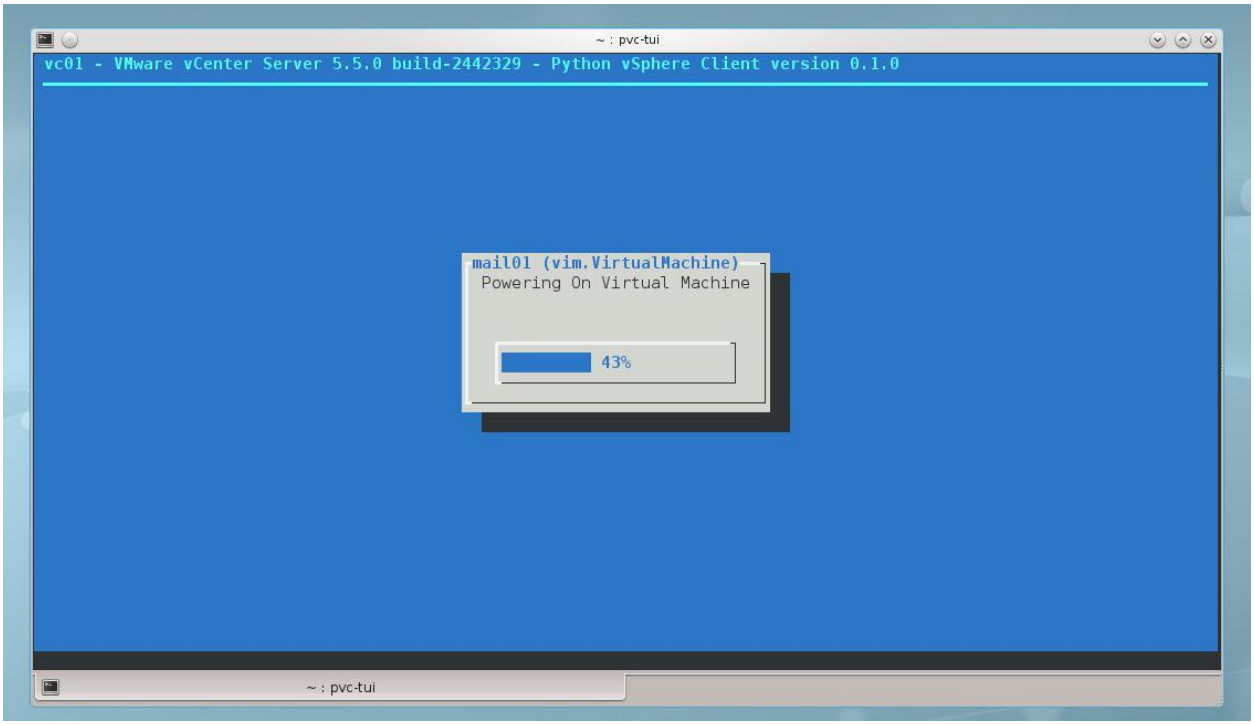



Figure 5.14: PVC displaying the progress of powering on a Virtual Machine

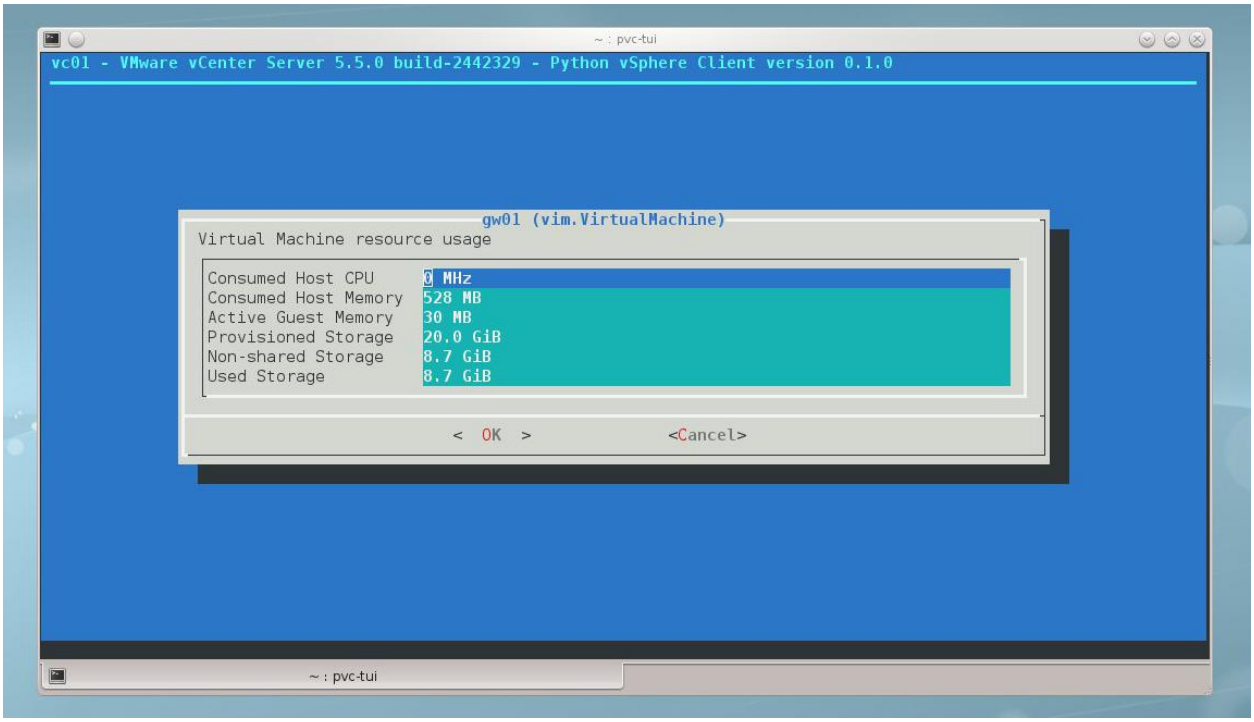



Figure 5.15: PVC displaying the resource usage of a Virtual Machine

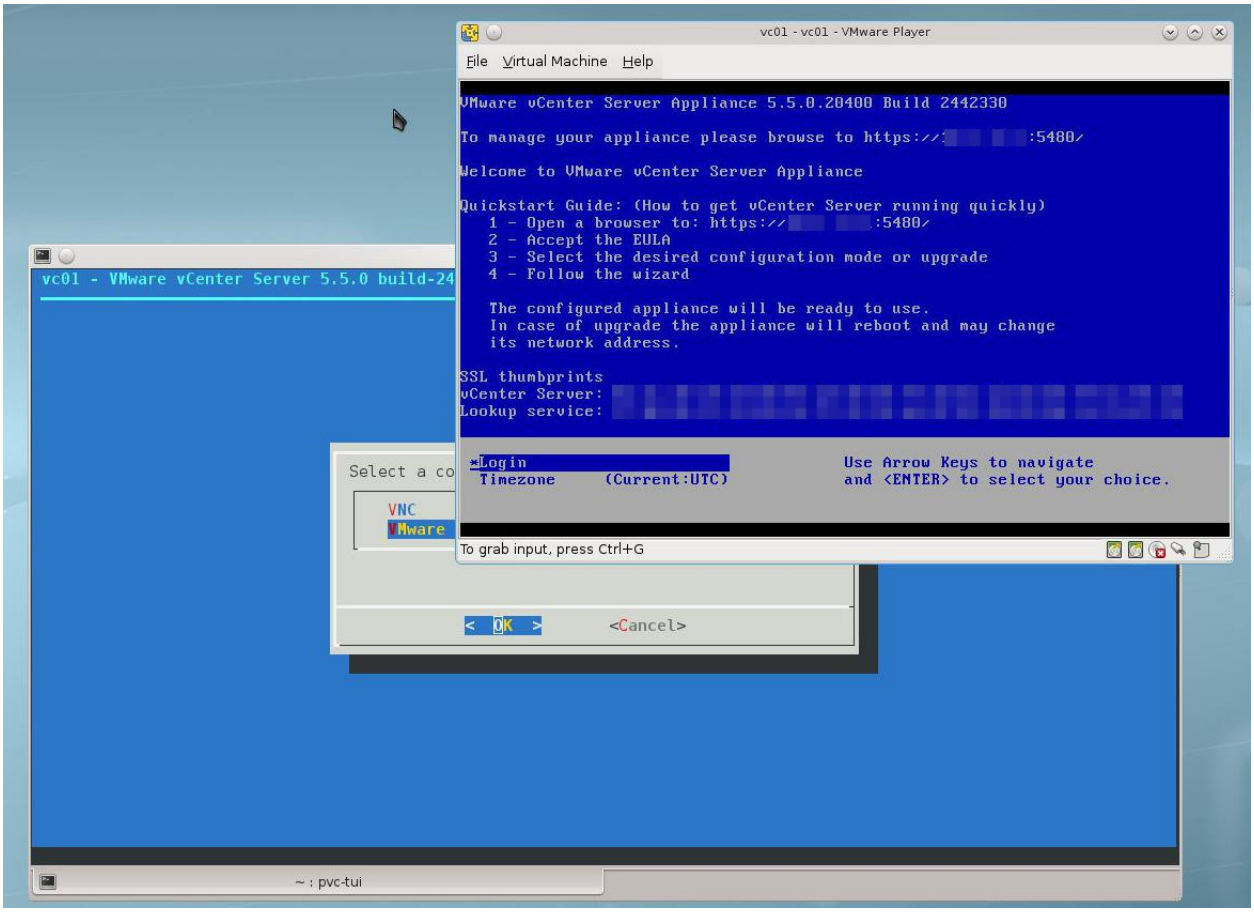



Figure 5.16: A screenshot showing PVC successfully launching a VMware Player console to a Virtual Machine

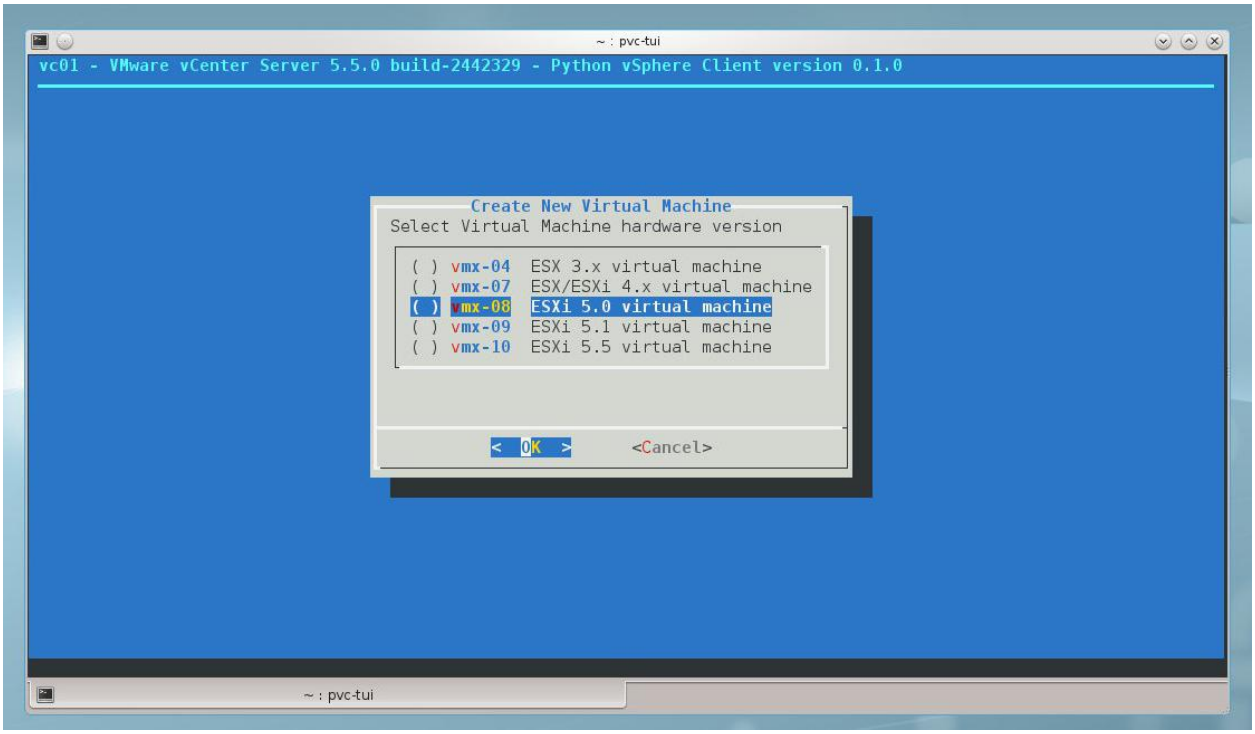



Figure 5.17: PVC displaying a menu of available hardware versions to choose from during creation of a new Virtual Machine.

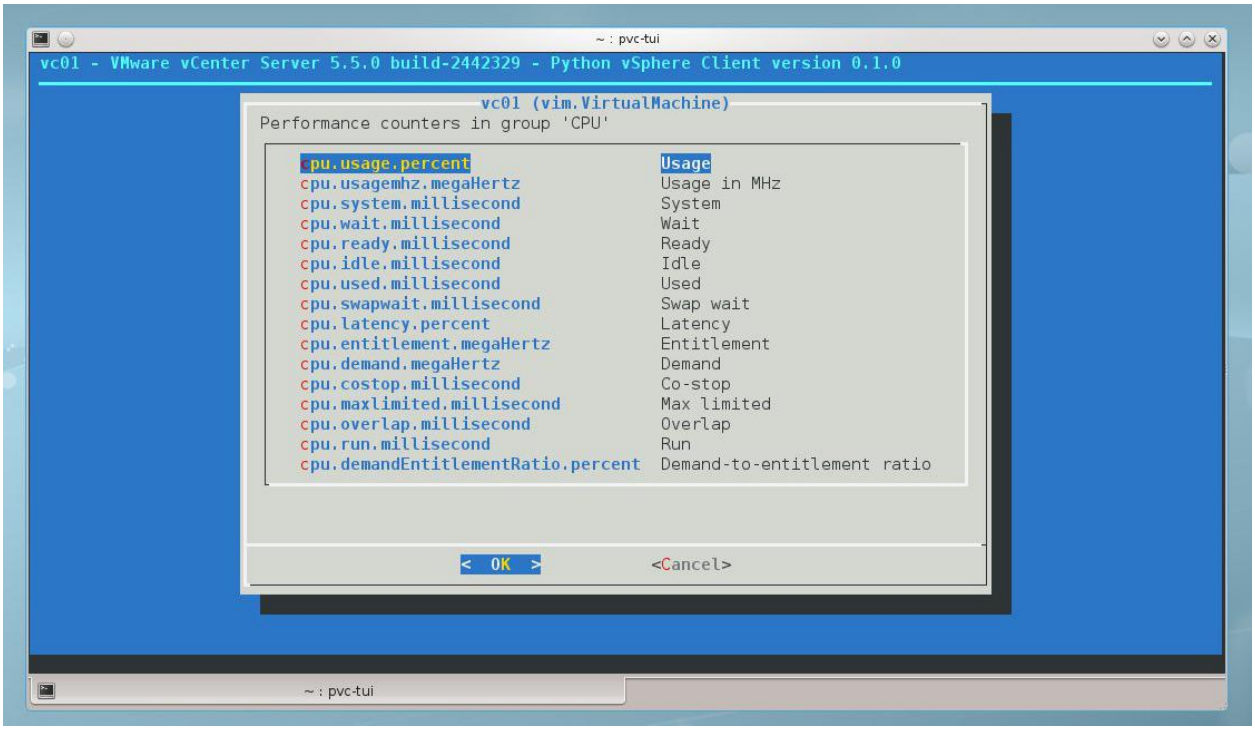



Figure 5.18: PVC displaying a menu of available real-time performance counters on a Virtual Machine