OCW VM - A Self Contained OCW Environment

The guide will walk you through the steps necessary to build a clean virtual environment for OCW work.

The login information for the VM is always **username: vagrant, password: vagrant.**

Requirements

You need to have Vagrant and one of Vagrant's supported virtual machine providers installed first. This guide (much like Vagrant's default guide) will assume that you're using VirtualBox.

Building

To start the VM build, change to the **ocw-vm** directory and run **vagrant up.**

```
cd $OCW_HOME/ocw-vm
vagrant up
```

Vagrant will now build your VM and provide you with a simple GUI. Login using the usual credentials (**username: vagrant, login: vagrant**) and install the necessary dependencies. After you log in you will need to copy the dependency installation file from /vagrant to your home directory and start the installation.

```
cp /vagrant/init-ocw-vm.sh .
./init-ocw-vm.sh
```

This process can take a while depending on the quality of your internet connection. Be patient, it will get there eventually.

> ⓘ **Help VM GUI TIps**
>
> When you're in the VM's GUI window you find that you're unable to use your mouse on our main desktop. To "escape" your mouse from the VM window press Left-Cmd (OS X) or Left-Ctrl (Windows/Linux). If your keybindings are different you may need to check the settings for additional information. Similarly, if the VM windows goes to sleep, you can wake it up by putting the window into focus and pressing the Escape key.

> ⚠ The VM build has switched over to using Miniconda instead of the full Anaconda. This helps speed up the build. The below screenshots may look slightly different given that, but all the inputs should be the same. If you run into problems please email the dev@ list.

About half way through the installation you will be prompted for input. The VM build relies on the Continuum Analytics Anaconda Python distribution which requires you to agree to their license and provide some minor configuration parameters.

```
○ ○ ○              ocw-vm_default_1400605262050_94035 [Running]
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
Cloning into 'climate'...
remote: Counting objects: 11532, done.
remote: Compressing objects: 100% (4220/4220), done.
Receiving objects: 100% (11532/11532), 3.65 MiB | 1.79 MiB/s, done.
remote: Total 11532 (delta 5429), reused 11156 (delta 5210)
Resolving deltas: 100% (5429/5429), done.

----------------------------------------------------------------------
                        Welcome to Easy OCW
----------------------------------------------------------------------


Checking for pip ...

Installing Anaconda Python distribution ...

*** NOTE *** When asked to update your PATH, you should respond YES.
Press [ENTER] to continue ...
 -  Downloading Anaconda ...
        done
 -  Installing ...

Welcome to Anaconda 1.9.2 (by Continuum Analytics, Inc.)

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
                              [icons]  [icons]  Left ⌘
```

You should answer **yes** to their license and use the default answers for the other prompts. After the install is complete, you will be prompted to add the new Anaconda path to your shell's configuration file. Answer **yes** to this as well. This step is important! If you forget to do this your build will not be able to find the Python dependencies that you installed!

After the installation has finished, restart the VM by going to the terminal where you ran **vagrant up** and run **vagrant halt**. We need to restart the VM so the GUI that we installed will kick in. Restart it by running **vagrant up** again.

```
vagrant halt
# Wait for the VM to shutdown
vagrant up
```

You should now be greeted with a splash screen asking you to log in. Select the vagrant user and log in with **vagrant** as the password.

Note, if your login simply redirects you back to the login page, click the small Ubuntu icon by the username and select the **Xfce Session**.
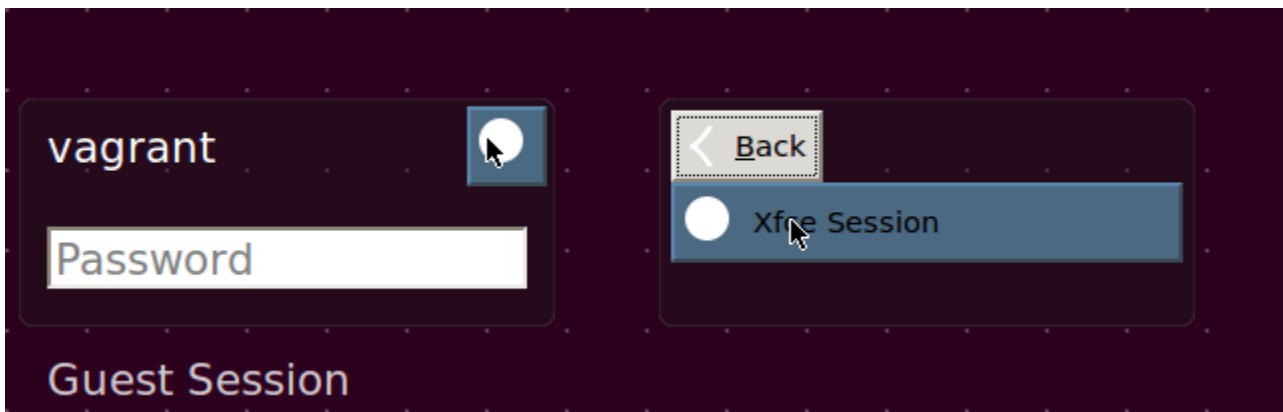
When prompted, select to use the Default Config

Congratulations. The VM is now configured!

Testing the Build

Open a Terminal by clicking **Applications Menu > Terminal Emulator**. Then run the following:

```
cd climate/examples
python simple_model_to_model_bias.py
```

Once the evaluation finished, you should find a **.png** in the examples directory if everything ran successfully.

Exporting the Virtual Machine from VirtualBox

Ensure that the VM is shutdown by running **vagrant halt** or closing the GUI and selecting the Shut Down option.

Open VirtualBox. You should see a VM with a name similar to **ocw-vm_default_1400605262050_94035.** You can rename in the settings for the VM.

Delete the default shared folder by **Right Clicking the VM > Settings > Shared Folders > Select the vagrant folder and select the delete icon (red minus sign on a folder)**

From VirtualBoxes settings, select the **Export Appliance.** Select the VM and click Continue. Select somewhere to save it and click Continue, then click Export.

Once this is done, your VM is ready!

Adding a Shared Folder

If you're received an exported version of the OCW VM, you may need to update the shared folder settings so you can access local data on your computer inside of the VM. This section assumes that you're using VirtualBox to host your VMs. If you're not, you'll need to check the documentation for the application that you're using.
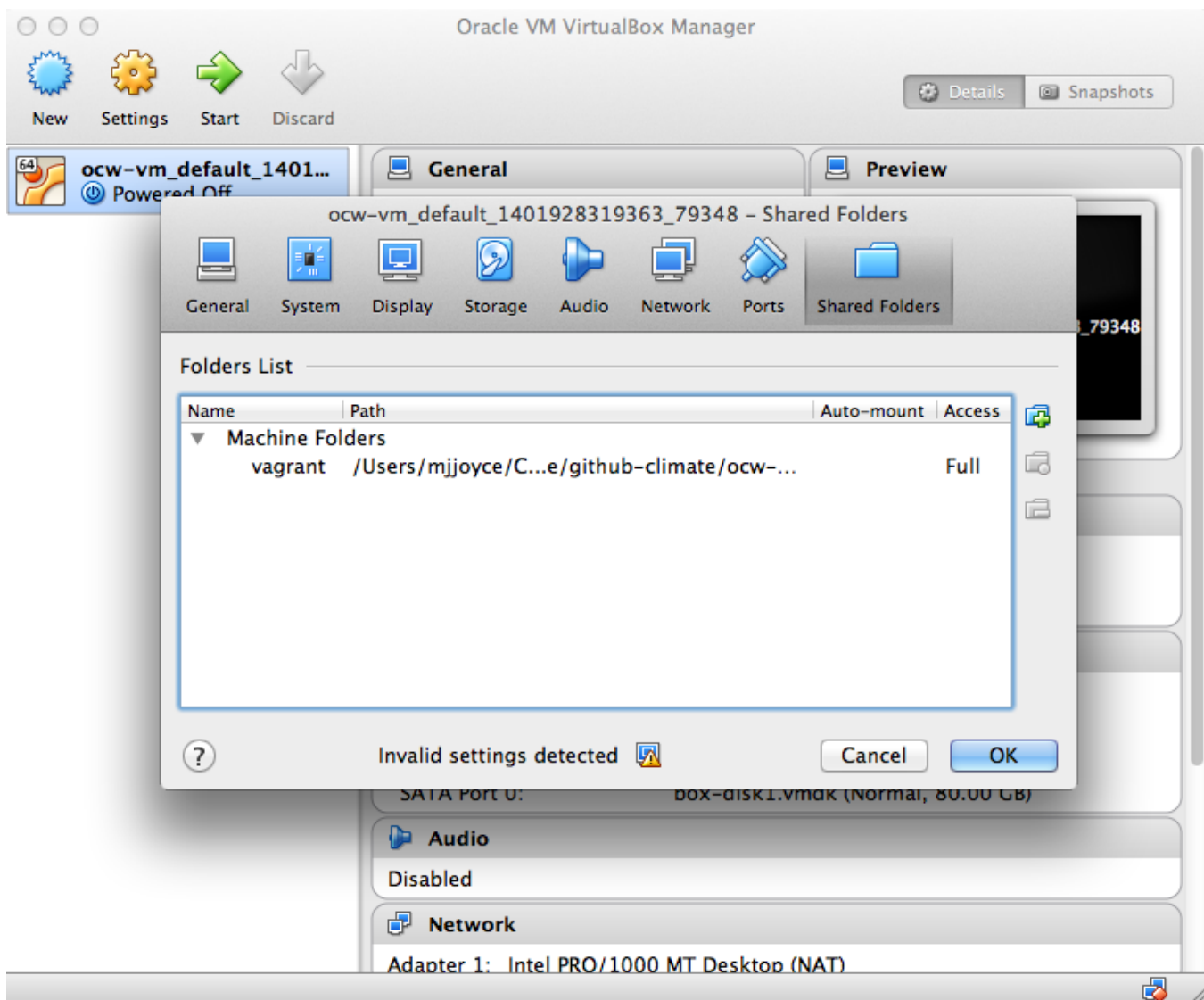
You should see something similar to the below image when you've imported your VM.
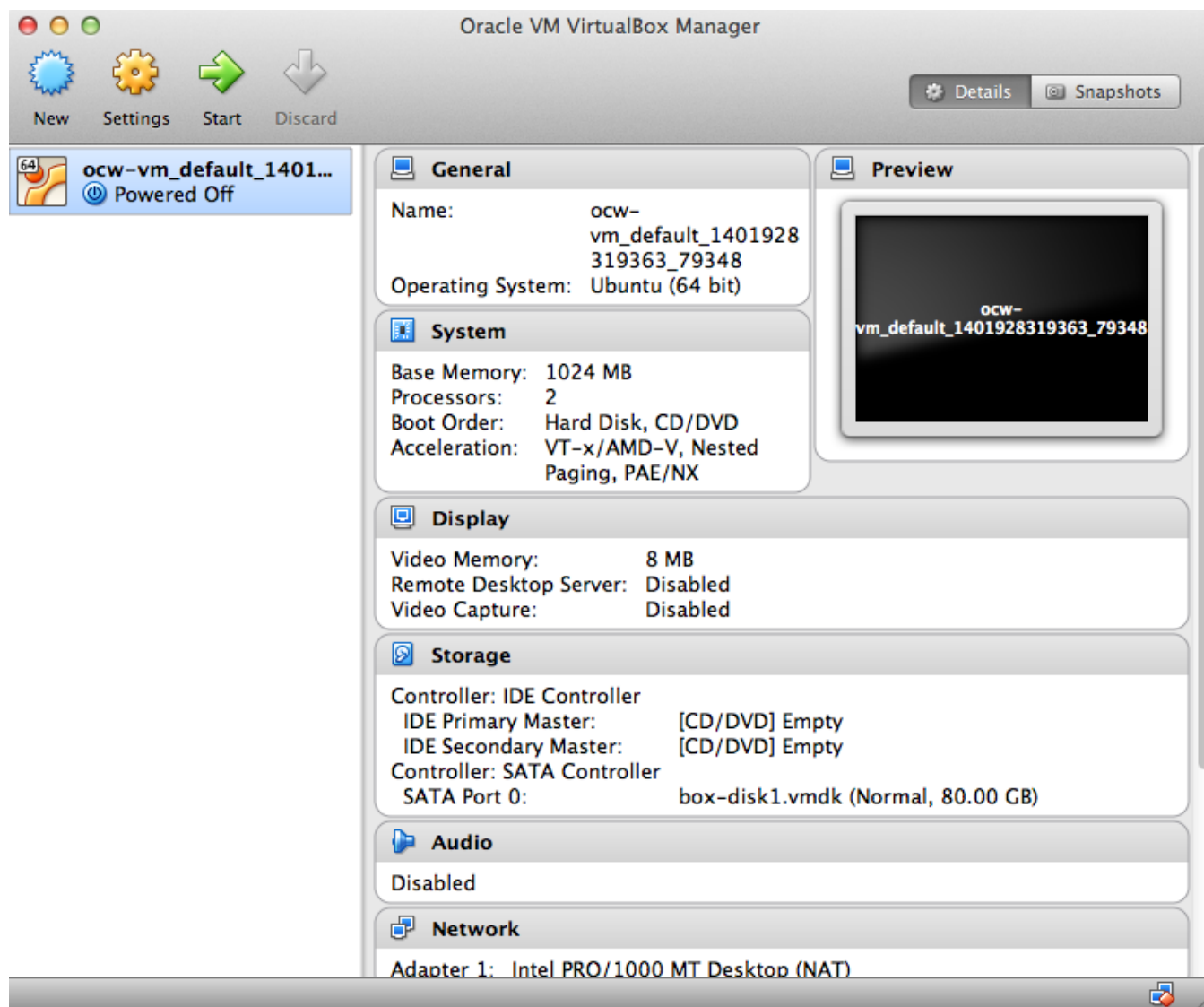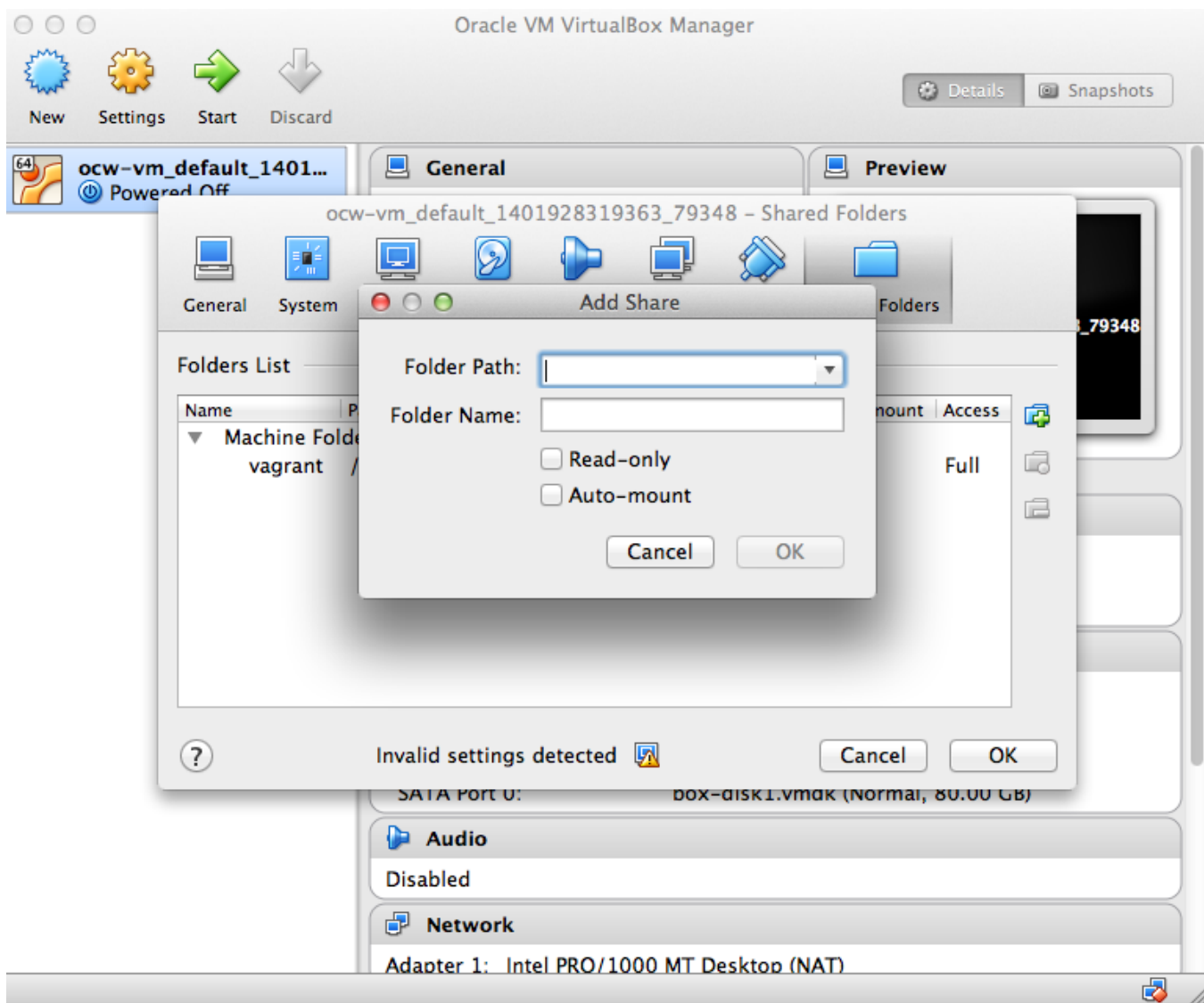

Right click the VM and select **Settings**.
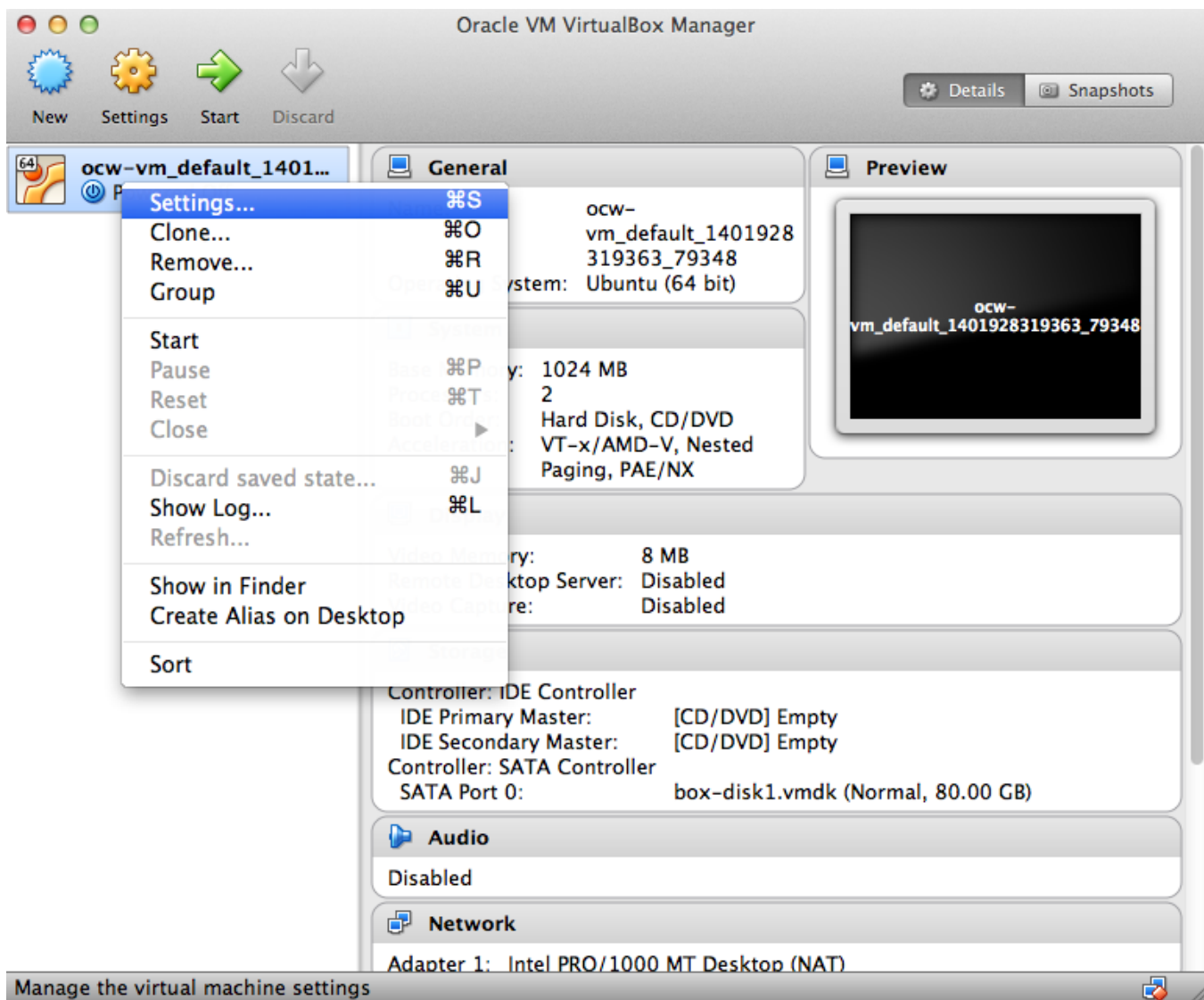
Select the **Shared Folders** Tab.

Most likely you folder list will be empty. Add a new one by clicking the icon with a "+" on a folder. You should see the following window pop up.

Navigate to the desired folder and give it a name. You'll most likely want to leave **Read-only** unselected so you can read and write to the directory from the VM. You'll also want to select **Auto-mount** so you don't have to manually mount the folder in your VM every time you start it up.