

Recomputation @ CP2015

User Guide

1. Purpose

This document provides instructions to those who wish to create Virtual Machines (VMs) for recomputation of experiments using the recomputation.org website.

2. Prerequisites

The following is required prior to using the recomputation.org website.

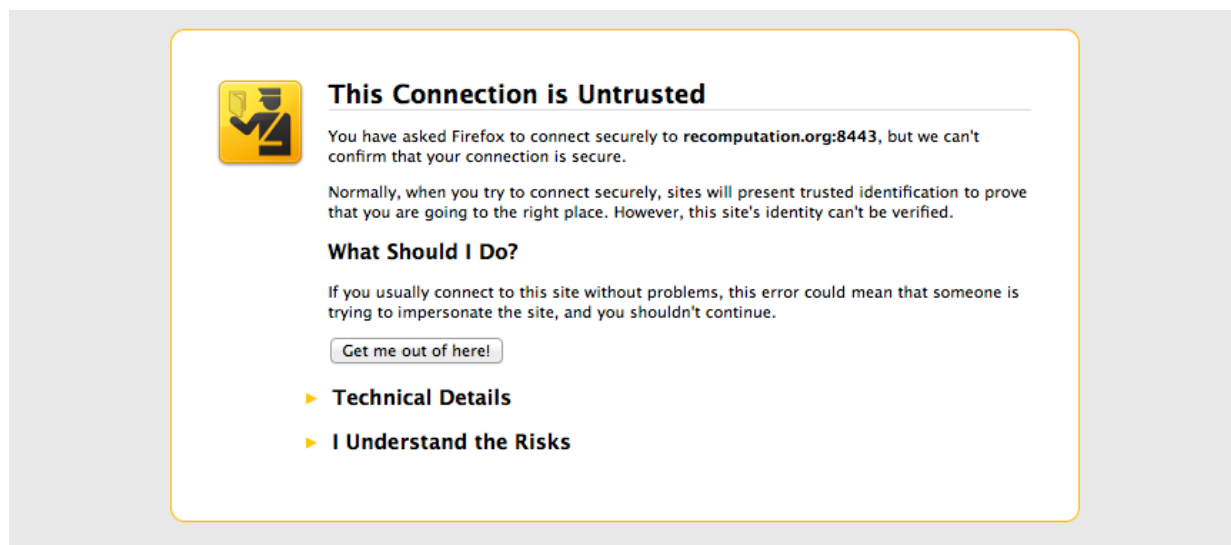
1. An account at [GitHub](https://github.com) (<https://github.com>). The recomputation.org website does not manage users and therefore relies on GitHub's user authentication mechanism to validate registered users. However, this website does not have access to your GitHub information besides any public data associated with your account.
2. A basic knowledge of Linux. The VMs generated at this website are based on Ubuntu, variant (i.e. distribution) of Linux.

3. Signing in to the recomputation.org website

The website is at present located at the following URL. You may click on the link below or copy/paste the URL on to your web browser.

<https://recomputation.org/recomputator>

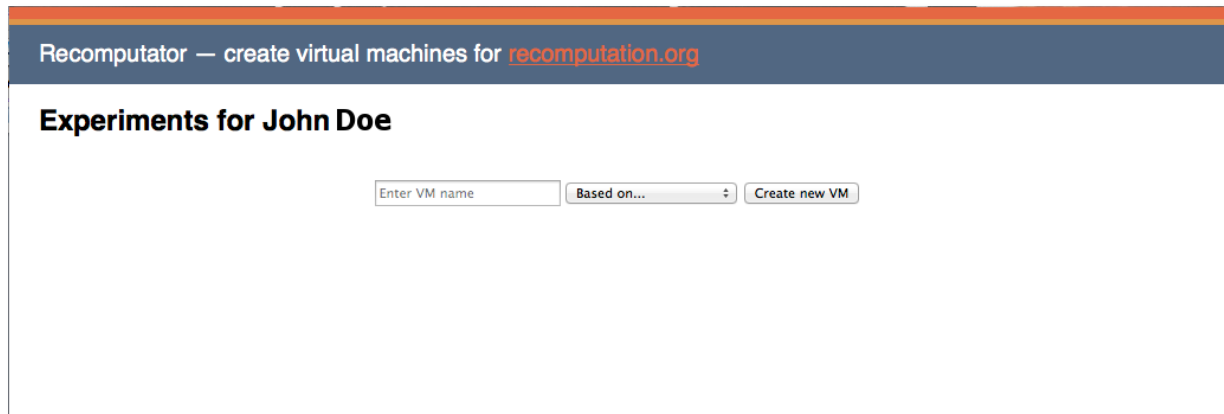
This website is work in progress and therefore currently uses a self-signed certificate for encrypting information. Your web browser will show a warning similar to the following as self-signed certificates are considered a security risk. However in this case you can safely ignore the warning and instruct the browser to proceed with displaying the page.



The browser will next take you to the login page at GitHub where you will need to enter the login name and password of your GitHub account. Note that this information is completely transparent to the recomputation.org website. GitHub will also prompt you to grant permission to "Recomputation" to access public data in your GitHub account.

4. Creating a VM for Recomputing an Experiment

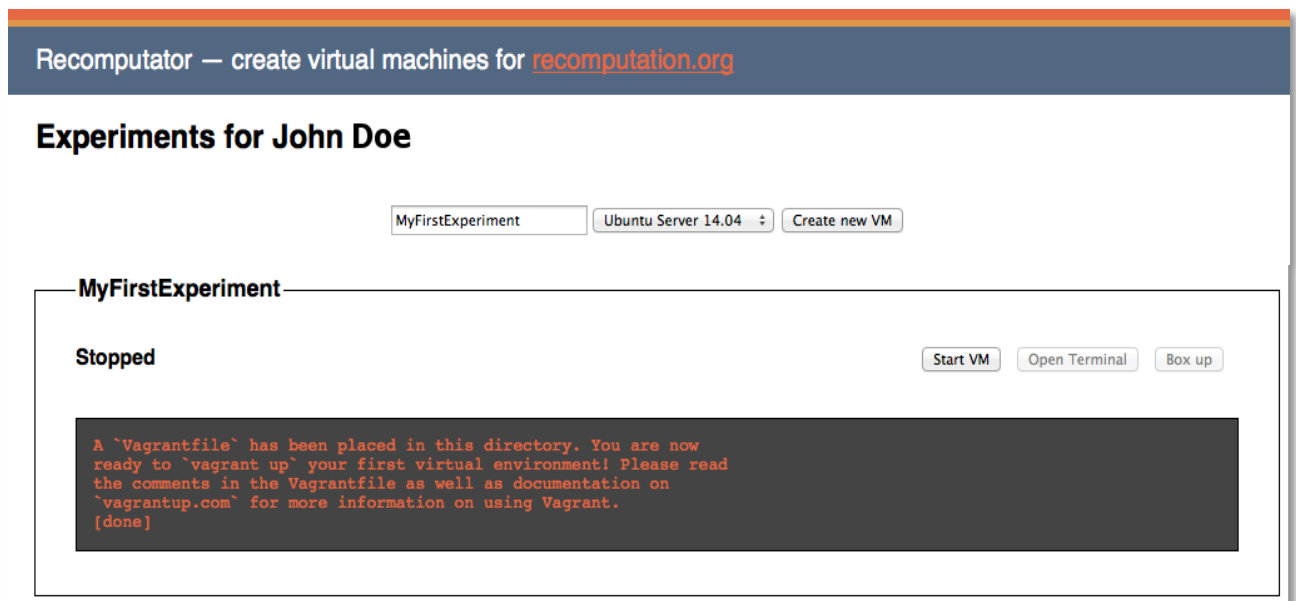
After you have successfully authenticated using your GitHub account, the landing page of your account at recomputation.org will be shown to you. As you have not yet created a VM, the landing page should look as follows.



The screenshot shows the top of the Recomputator interface. At the top is a dark blue header with the text "Recomputator — create virtual machines for recomputation.org". Below this is a white section titled "Experiments for John Doe". Inside this section, there are three input fields: "Enter VM name", "Based on..." with a dropdown arrow, and a "Create new VM" button.

The steps for creating a VM are presented next.

1. Type the name of the VM you are about to create in the "Enter VM name" field, for example, "MyFirstExperiment". This is similar to naming an ordinary file and will help you identify a specific VM or experiment from a collection of VMs.
2. Select the base operating system image for the new VM from the dropdown list named "Based on...". Initially this list contains three images which are barebones versions of 64-bit Ubuntu distributions, named according to version. However, as you create your own VMs, they will be added to this list, allowing you to have a customised base image that can be reused for creating other VMs to run your experiments.
3. Click on the "Create new VM" button. This sets up the necessary configuration files for creating your new VM. However, the VM itself is not yet created and will be created when you boot the new VM for the first time. The browser should display a screen similar to the following.



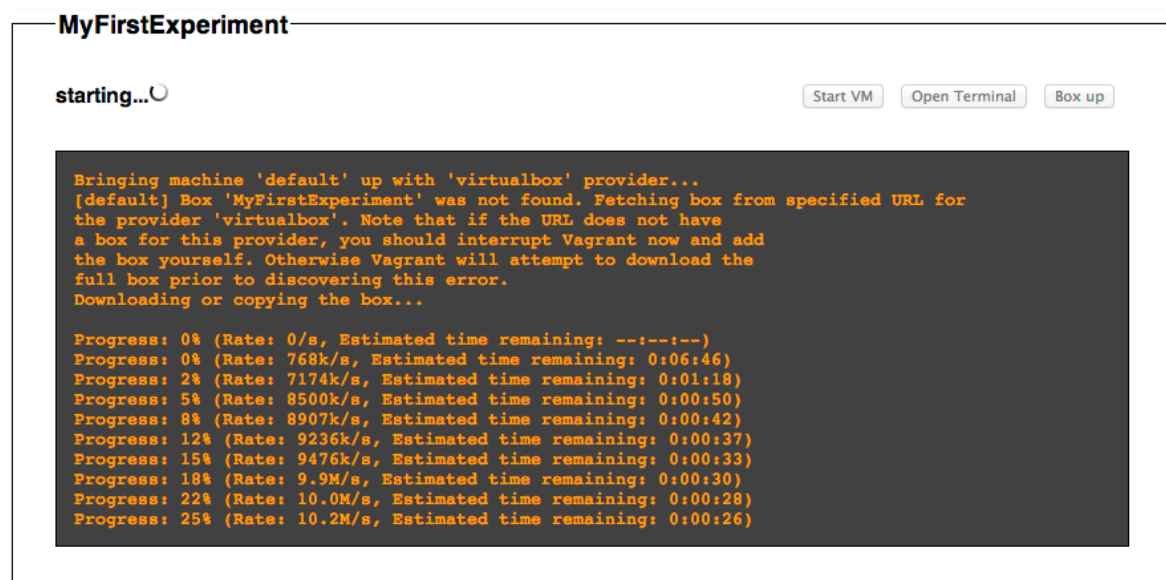
The screenshot shows the Recomputator interface after a VM has been created. The header is the same. The "Experiments for John Doe" section now shows the VM name "MyFirstExperiment" in the input field, "Ubuntu Server 14.04" in the "Based on..." dropdown, and the "Create new VM" button. Below this, there is a section titled "MyFirstExperiment" with a sub-header "Stopped". To the right of "Stopped" are three buttons: "Start VM", "Open Terminal", and "Box up". Below these buttons is a dark grey terminal window with the following text: "A `Vagrantfile` has been placed in this directory. You are now ready to `vagrant up` your first virtual environment! Please read the comments in the Vagrantfile as well as documentation on `vagrantup.com` for more information on using Vagrant. [done]"

5. Configuring a VM and Deploying Experiments

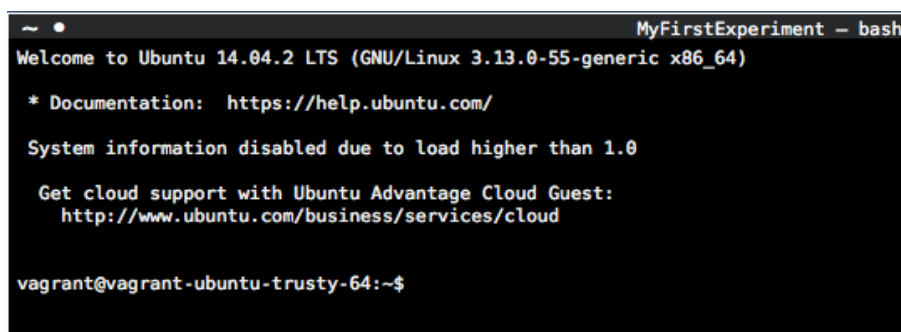
Once a VM has been created, the next step is to boot the VM and login to the system, as you would do with a real computer. This step does not have to immediately follow the VM creation step. Instead, you may visit the recomputation.org/recomputator website at any time to configure and work with VMs you created earlier. The VMs you create are stored in our servers indefinitely unless otherwise instructed to remove them.

The following steps for booting and working with a VM are the same whether this activity is done immediately upon VM creation or at a latter stage.

1. Click on the “Start VM” button associated to VM with which you wish to work. If this is the first time you are starting a particular VM, it will also create the resources for the VM prior to booting it up. This process should take a few minutes (or sooner if the VM has already been created) during which time the page should display the progress similarly to what is illustrated below. You will also notice that the buttons are greyed out until the VM booting process is complete. Once complete these buttons will be re-enabled and the “Start VM” button re-named to “Stop VM”. At this point you have successfully booted a VM that is now ready be configured for experiments.



2. Click on the “Open Terminal” button to open a terminal window through which you can login to the Ubuntu VM as you would with a normal Linux based computer. The VM login process does not require you to provide a user name and a password, but instead employs preconfigured user credentials to log you in automatically. Once you are connected to the VM, you should see the familiar Linux command prompt. This VM is a full-functional Linux computer, which should be easily configurable to deploy and execute your experiments. The terminal window of the VM looks like this.



3. Use the terminal window to configure the Linux VM as required prior to deploying your experiments. For example, you could use the Linux package manager **apt-get** to download and install any packages that are not available in the VM.
4. Deploy your experiment. Note that every VM created at recomputation.org has access to the Internet. Therefore you may download files related to your experiment using standard Linux tools such as **wget**. Furthermore, it is recommended that the software and data required for the experiment be installed within the **/home/vagrant** directory of the VM. This is the home directory of the preconfigured user.
5. Once the necessary components of your experiment have been deployed and correctly configured, make sure to test your experiment in the VM environment. You shouldn't treat the VM any different from an ordinary Linux based computer during these tests. If the VM needs to be rebooted, use the "Stop VM" button to shutdown the VM and then the "Start VM" button to restart system.
6. Unless explicitly shutdown using the "Stop VM" button, a VM that has been started will continue to run on the recomputation.org servers even after you have closed the web browser used for interacting with that VM. You may return to the website at anytime to execute new experiments or check the status of a long running experiment.

Note: Users are strongly discouraged from keeping their VMs running unnecessarily over long periods, as active VMs consume a lot of the resources of the physical servers that host them. Therefore, please be mindful of this fact so that every user of recomputation.org has sufficient resources to execute their experiments.

6. Boxing up a VM

The VMs you create and setup with experiments are initially stored in the servers of recomputation.org. While the website itself is publicly accessible, users are able to access only their own VMs. Therefore, in order to make your VMs available to a wider audience, the system provides a facility to download your VMs to the local computer. This process is termed *boxing up* a VM, as the VM is packaged or "boxed" into a single file that can be easily distributed. To host these VMs in your local computer (and to execute experiments), they require two pieces of free software. First is VirtualBox, a virtualization product and the second is Vagrant, which works closely with VirtualBox to automate some of its operations. Make sure both these software tools are installed in the computer where you wish to deploy boxed VMs downloaded from this website.

The steps involved in boxing up a VM are presented below.

1. Click on the "Box up" button associated with the VM you wish to download. If the VM is already running, it is shutdown before boxing up. After a few minutes, a link saying "Download package" is shown, indicating that the boxed VM is ready to be downloaded.



2. Click on the “Download package” link to download the VM to your local computer. Note that if you make any changes to a VM hosted our servers after it has been boxed up, those changes are not included in the boxed version. Therefore, please make sure to box up the VM again if you wish to download the updated VM.

7. Deploying a boxed VM in the Local Computer

Once a boxed VM has been downloaded from the recomputation.org website, follow these steps to deploy the VM in the local computer.

1. Install **VirtualBox**. This free software can be downloaded from: <https://www.virtualbox.org>
2. Install **Vagrant**, which can be downloaded from: <http://www.vagrantup.com>
3. Open a terminal / console / command line window.
4. Go to the directory where the downloaded VM is saved, and type the following command. Make sure to replace `<boxed_vm_name>` with actual name of the downloaded .box file.

```
vagrant init <boxed_vm_name>
```

This command will create file named Vagrantfile in the same directory. This file will together with the boxed VM image are used for creating the VM in VirtualBox.

5. To create the VM in VirtualBox and to fire it up, issue the following command.

```
vagrant up
```

6. At this point the VM should be running in your local computer, hosted by VirtualBox. To sign in to the VM, give following command.

```
vagrant ssh
```

You should now be at the command prompt of the VM and able to execute your experiments. To return to local computer type *exit* or *Ctrl-D*.

7. You may wish to turn off the VM when it's no longer in use as running VMs consume memory and processor bandwidth of the physical computer. To shutdown the VM issue the command:

```
vagrant halt
```

To restart the VM you may issue the command *vagrant up* at any time.

8. If required, you could completely remove the VirtualBox VM image stored in your local hard drive. To do so, issue the following command.

```
vagrant destroy
```