# X-forwarding to run GUI program in Vagrant box

Vagrant (/search/Vagrant) VirtualBox (/search/VirtualBox) X-Server (/search/X-Server) GUI (/search/GUI)

How can run a desktop GUI application inside my headless VirtualBox (https://www.virtualbox.org/) that was launched via Vagrant (https://www.vagrantup.com/) ?

I already had an explanation on how to set up Vagrant with VirtualBox (/setting-up-vagrant). I assume you already have all that set up.

There are a

## X-Server on the host

You need to have an X-Server on you host-machine.

If you run a desktop Linux system as your host as well, then you already have an X Server.

On Mac OSX you can install XQuartz (https://www.xquartz.org/)

When writing this article I have not tried it on MS Windows, but as I recall I used Xming (https://freedesktop.org/wiki/Xming/) at one of my clients.

## Enable X-forwarding

You need to enable X-forwarding in the guest operating system. Probably the best is to do it via the Vagrant configuration file `Vagrantfile` :

```
config.ssh.forward_x11 = true
```

## ssh into the box

Instead of using `vagrant ssh` we'll need to use the `ssh` command to access the guest operating system. Run `vagrant ssh-config` on the host in order to find out the configuration details.

```
$ vagrant ssh-config

Host default
  HostName 127.0.0.1
  User ubuntu
  Port 2222
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /Users/gabor/work/.vagrant/private_key
  IdentitiesOnly yes
  LogLevel FATAL
  ForwardX11 yes
```

From this we can get the **User**, the **HostName**, the **Port**, and the location of the **IdentityFile** that holds the private key we need to use.

In addition we need to supply the `-X` flag that tells ssh to use the X-forwarding.

```
ssh ubuntu@127.0.0.1 -p 2222 -i /Users/gabor/work/.vagrant/private_key -X
```

Then you can already start desktop GUI applications.

Traditionally `xclock` and `xeyes` were used to test this as they are really simple X-based applications, but if you cannot install either of those, you might have something like `xarclock`.

Once you know you can launch x applications in the guest and see them on the host, I recommend creating and alias for the command by adding this to your `~/.bashrc` or `~/.bash_profile` in your host. (Assuming Linux or OSX)

```
alias vssh='ssh ubuntu@127.0.0.1 -p 2222 -i /Users/gabor/work/.vagrant/private_k
```

and the reloading it using `source`.

That way the connection will be just a `vssh` away.

Gabor Szabo

(https://plus.google.com/102810219707784087582?rel=author)

## Comments

In the comments, please wrap your code snippets within <pre> </pre> tags and use spaces for indentation.

## Author: Gabor Szabo

Gabor who runs the Code Maven site helps companies set up **test automation**, CI/CD **Continuous Integration** and **Continuous Deployment** and other **DevOps** related systems. Gabor can help your team improve the development speed and reduce the risk of bugs.

He is also the author of a number of eBooks (https://leanpub.com/u/szabgab).

Contact Gabor (https://szabgab.com/contact.html) if you'd like to hire his services.

If you would like to support his freely available work, you can do it via Patreon (https://www.patreon.com/szabgab).

(http://perlmaven.com/digitalocean)