61

How to Install and Configure VNC on Ubuntu 14.04

PostedOctober 23, 2014 731.9kviews [MISCELLANEOUS](https://www.digitalocean.com/community/tags/miscellaneous?type=tutorials) [UBUNTU](https://www.digitalocean.com/community/tags/ubuntu?type=tutorials)

**Introduction**

VNC, or "Virtual Network Computing", is a connection system that allows you to use your keyboard and mouse to interact with a graphical desktop environment on a remote server. VNC makes managing files, software, and settings on a remote server easier for users who are not yet comfortable with working with the command line.

In this guide, we will be setting up VNC on an Ubuntu 14.04 server and connecting to it securely through an SSH tunnel. The VNC server we will be using is TightVNC, a fast and lightweight remote control package. This choice will ensure that our VNC connection will be smooth and stable even on slower Internet connections.

Prerequisites

Before you begin with this guide, there are a few steps that need to be completed first.

You will need an Ubuntu 14.04 server installed and configured with a non-root user that has sudoprivileges. If you haven't done this yet, you can run through steps 1-4 in the [Ubuntu 14.04 initial server setup guide](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-14-04) to create this account.

Once you have your non-root user, you can use it to SSH into your Ubuntu server and continue with the installation of your VNC server.

Step One — Install Desktop Environment and VNC Server

By default, most Linux server installations will not come with a graphical desktop environment. If this is the case, we'll need to begin by installing one that we can work with. In this example, we will install XFCE4, which is very lightweight while still being familiar to most users.

We can get the XFCE packages, along with the package for TightVNC, directly from Ubuntu's software repositories using apt:

sudo apt-get update

sudo apt-get install xfce4 xfce4-goodies tightvncserver

To complete the VNC server's initial configuration, use the vncserver command to set up a secure password:

vncserver

(After you set up your access password, you will be asked if you would like to enter a view-only password. Users who log in with the view-only password will not be able to control the VNC instance with their mouse or keyboard. This is a helpful option if you want to demonstrate something to other people using your VNC server.)

vncserver completes the installation of VNC by creating default configuration files and connection information for our server to use. With these packages installed, you are ready to configure your VNC server and graphical desktop.

Step Two — Configure VNC Server

First, we need to tell our VNC server what commands to perform when it starts up. These commands are located in a configuration file called xstartup. Our VNC server has an xstartup file preloaded already, but we need to use some different commands for our XFCE desktop.

When VNC is first set up, it launches a default server instance on port 5901. This port is called a display port, and is referred to by VNC as :1. VNC can launch multiple instances on other display ports, like :2, :3, etc. When working with VNC servers, remember that :X is a display port that refers to 5900+X.

Since we are going to be changing how our VNC servers are configured, we'll need to first stop the VNC server instance that is running on port 5901:

vncserver -kill :1

Before we begin configuring our new xstartup file, let's back up the original in case we need it later:

mv ~/.vnc/xstartup ~/.vnc/xstartup.bak

Now we can open a new xstartup file with nano:

nano ~/.vnc/xstartup

Insert these commands into the file so that they are performed automatically whenever you start or restart your VNC server:

#!/bin/bash

xrdb $HOME/.Xresources

startxfce4 &

The first command in the file, xrdb $HOME/.Xresources, tells VNC's GUI framework to read the server user's .Xresources file. .Xresources is where a user can make changes to certain settings of the graphical desktop, like terminal colors, cursor themes, and font rendering.

The second command simply tells the server to launch XFCE, which is where you will find all of the graphical software that you need to comfortably manage your server.

To ensure that the VNC server will be able to use this new startup file properly, we'll need to grant executable privileges to it:

sudo chmod +x ~/.vnc/xstartup

Step Three — Create a VNC Service File

To easily control our new VNC server, we should set it up as an Ubuntu service. This will allow us to start, stop, and restart our VNC server as needed.

First, open a new service file in /etc/init.d with nano:

sudo nano /etc/init.d/vncserver

The first block of data will be where we declare some common settings that VNC will be referring to a lot, like our username and the display resolution.

#!/bin/bash

PATH="$PATH:/usr/bin/"

export USER="user"

DISPLAY="1"

DEPTH="16"

GEOMETRY="1024x768"

OPTIONS="-depth ${DEPTH} -geometry ${GEOMETRY} :${DISPLAY} -localhost"

. /lib/lsb/init-functions

Be sure to replace user with the non-root user that you have set up, and change 1024x768 if you want to use another screen resolution for your virtual display.

Next, we can start inserting the command instructions that will allow us to manage the new service. The following block binds the command needed to start a VNC server, and feedback that it is being started, to the command keyword start.

case "$1" in

start)

log\_action\_begin\_msg "Starting vncserver for user '${USER}' on localhost:${DISPLAY}"

su ${USER} -c "/usr/bin/vncserver ${OPTIONS}"

;;

The next block creates the command keyword stop, which will immediately kill an existing VNC server instance.

stop)

log\_action\_begin\_msg "Stopping vncserver for user '${USER}' on localhost:${DISPLAY}"

su ${USER} -c "/usr/bin/vncserver -kill :${DISPLAY}"

;;

The final block is for the command keyword restart, which is simply the two previous commands (stopand start) combined into one command.

restart)

$0 stop

$0 start

;;

esac

exit 0

Once all of those blocks are in your service script, you can save and close that file. Make this service script executable, so that you can use the commands that you just set up:

sudo chmod +x /etc/init.d/vncserver

Now try using the service and command to start a new VNC server instance:

sudo service vncserver start

Step Four — Connect to Your VNC Desktop

To test your VNC server, you'll need to use a client that supports VNC connections over SSH tunnels. If you are using Windows, you could use TightVNC, RealVNC, or UltraVNC. Mac OS X users can use the built-in Screen Sharing, or can use a cross-platform app like RealVNC.

First, we need to create an SSH connection on your local computer that securely forwards to the localhost connection for VNC. You can do this via the terminal on Linux or OS X via the following command:

(Remember to replace user and server\_ip\_address with the username and IP you used to connect to your server via SSH.)

ssh -L 5901:127.0.0.1:5901 -N -f -l user server\_ip\_address

If you are using a graphical SSH client, like PuTTY, use server\_ip\_address as the connection IP, and set localhost:5901 as a new forwarded port in the program's SSH tunnel settings.

Next, you can use your VNC viewer to connect to the VNC server at localhost:5901. Make sure you don't forget that :5901 at the end, as that is the only port that the VNC instance is accessible from.

Once you are connected, you should see the default XFCE desktop ready for configuration and use! It should look something like this:

First VNC<br>
connection

Once you have verified that the VNC connection is working, add your VNC service to the default services, so that it will automatically start whenever you boot your server:

sudo update-rc.d vncserver defaults

Conclusion

You should now have a secured VNC server up and running on your Ubuntu 14.04 server. Now you'll be able to manage your server's files, software, and settings with an easy-to-use graphical interface.