Installation and connection to a Desktop Environment on a remote VM

Simone Garau
20005068@studenti.uniupo.it

1. Goal

In this guide, we will cover how to install a desktop environment (DE) on an Ubuntu virtual machine (VM) and configure TigerVNC to control the machine remotely via an SSH connection. We will explore how to install and configure different Ubuntu flavors: Kubuntu, Xubuntu, Lubuntu, and classical Ubuntu. Additionally, we'll explain the components of a desktop environment on Linux and what is a VNC service in Section 6. It's important to note that while a VNC server can enable various methods of machine control, our focus here is specifically on remote graphical interface access.

2. Prerequisites

- An Ubuntu virtual machine (VM) to any provider as long as you can access it.
- An Ubuntu client for remote connection.
- Root or sudo privileges on both machines.

Technically speaking, you can follow this guide even if you run different versions of GNU Linux like, for instance, Fedora or Arch but you have to consider that some parts might be different and others can be really apart and tricky. Especially on the VM! So forewarned is forearmed.

3. On Virtual Machine

Usually a VM on any cloud provider doesn't have a Desktop Environment (DE) installed and we can interact with the classical Command Line Interface (CLI), sometimes it seems just old and uncomfortable.

3.1. Installing the DE

Fist of all you have to update your system with the following command:

```
sudo apt-get update && sudo apt-get upgrade -y
```

After that we can choose our favorite flavor and use the specific command, choose from the table below.
Note that we're going to see some of the options, non all flavors.

3.1.1. Choosing the Desktop Environment

Flavours	
GNOME (Ubuntu)	`sudo apt install ubuntu-gnome-desktop`
KDE Plasma (Kubuntu)	`sudo apt install kubuntu-desktop`
XFCE (Xubuntu)	`sudo apt install xubuntu-desktop`
LXQt (Lubuntu)	`sudo apt install lubuntu-desktop`

It's highly possible that you can be prompted to choose a display manager (DM), choose the one needed for the DE that you chose previously.

3.1.2. Choosing the Display Manager

Display Manager	
GNOME (Ubuntu)	gdm
KDE Plasma (Kubuntu)	sddm
XFCE (Xubuntu)	lightdm
LXQt (Lubuntu)	sddm

Move with the tab key, select an option with the spacebar and confirm the choice with enter.

3.1.3. Done!

You have installed the DE, but at the current state of work you cannot access it because the SSH connection from your system with bash doesn't allow it. We need a VNC Server.

3.2. Installing and Configuring TigerVNC

A VNC Server is a utility which starts a Virtual Network Computing (VNC) desktop. It runs Xvnc with appropriate options and starts a window manager on the VNC desktop. VNC server allows users to run separate sessions in parallel on a machine which can then be accessed by any number of clients from anywhere.
You can install TigerVNC with just a simple command because is a service on the official Ubuntu repository:

```
sudo apt install tigervnc-standalone-server tigervnc-common
```

It's mandatory to create the startup file that will make the DE available:

```
sudo nano ~/.vnc/xstartup
```

Nano is a text edit tool that works like you are in a bash, so you can paste the following code by the shortcut ctrl + shift + V. Here the code:

```
#!/bin/bash
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
export XKL_XMODMAP_DISABLE=1
startDE & #Replace with proper DE command
```

Of course if you like other tools it makes no difference.
The last line is dependent on the DE that you chose previously, use the table below to replace the last line.

Desktop Environment	
GNOME (Ubuntu)	`gnome-session &`
KDE Plasma (Kubuntu)	`startplasma-x11 &`
XFCE (Xubuntu)	`startxfce4 &`
LXQt (Lubuntu)	`startlxqt &`

Now that file has to be executable so run this command:

```
chmod +x ~/.vnc/xstartup
```

Without that when TigerVNC tries to start, it surely can't execute the proper commands.

3.2.1. Explanation of the xstartup file for GNOME:

- `#!/bin/bash`: Specifies that the file should be executed using Bash.
- `[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup`: If `/etc/vnc/xstartup` exists and is executable, execute it.
- `[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources`: If the `.Xresources` file exists and is readable, load it using `xrdb`. This file contains X resources configurations like font and color settings.
- `xsetroot -solid grey`: Sets the root window background color to grey. This is a temporary background before the full desktop environment starts.
- `export XKL_XMODMAP_DISABLE=1`: Disables X Keyboard Extension's XMODMAP to avoid keyboard mapping conflicts.
- `gnome-session &`: Starts the GNOME session.

3.2.2. Set the VNC password

Now it's mandatory to set the password in order to access from remote, so run this command:

```
vncserver
```

The service will ask you to set a password and to confirm by inserting the same password again. It will also ask for a read-only password that allows clients to see what's going on without interacting.

3.2.3. Set basics configurations

You can start the service with some properties specifying on the command some params but I suggest to make a file that will load all of your preferences:

```
sudo nano ~/.vnc/config
```

Now insert what you need like interface, port, display resolution etc. Of course you can avoid creating this file and inserting it manually.

```
rfbport=your_port
interface=your_interface_like_0.0.0.0
```

Replace with proper values. **The standard port is 5901 if not specified**.

3.2.4. Check the service

Now you can check if it is all setted up properly by running this command:

```
netstat -tuln | grep your_port
```

Replace ¨your_port¨ with your previous choice or with the default 5901. If the service is running you can see something like that:

In my case the port was the default one.

Troubleshooting

If none of that appended you can try to reboot the machine and try again. After that if you are still unable to see the service active you probably need to work with *systemd* which is the system manager that allows us to work with services. Try to start the service with this command:

```
systemctl start vncserver@:1.service
```

If it works, make sure this service will start automatically with the system by execute this command:

```
systemctl enable vncserver@:1.service
```

Technically that ¨@:1¨ can have other numbers due to the possibility to have multiple shared displays but, we assume you just want to share a simple interface, that's why the one.

4. On Client

On the client we need the viewer that allows us to access to the service on the remote VM, so install it by run this command:

```
sudo apt install tigervnc-viewer
```

The viewer doesn't need any configuration and is ready to use.

4.1. SSH Connection

We need to create a tunnel connection with the server so run this ssh command:

```
ssh -L local_port:localhost:vm_port -N -f -l username VM_IP
```

Replace as it comes:

local_port	A free port on your local machine, it can be any number as long as it's free.

vm_port	The port you chose during the VM configuration
username	The username you use to access the VM
VM_IP	The VM's public IP

Assuming it is all correct you will not be prompted after the execution of this command, otherwise you'll see some error messages depending on the error: wrong username, busy port etc.

4.2. Start the Viewer

Just run the following command and see the magic:

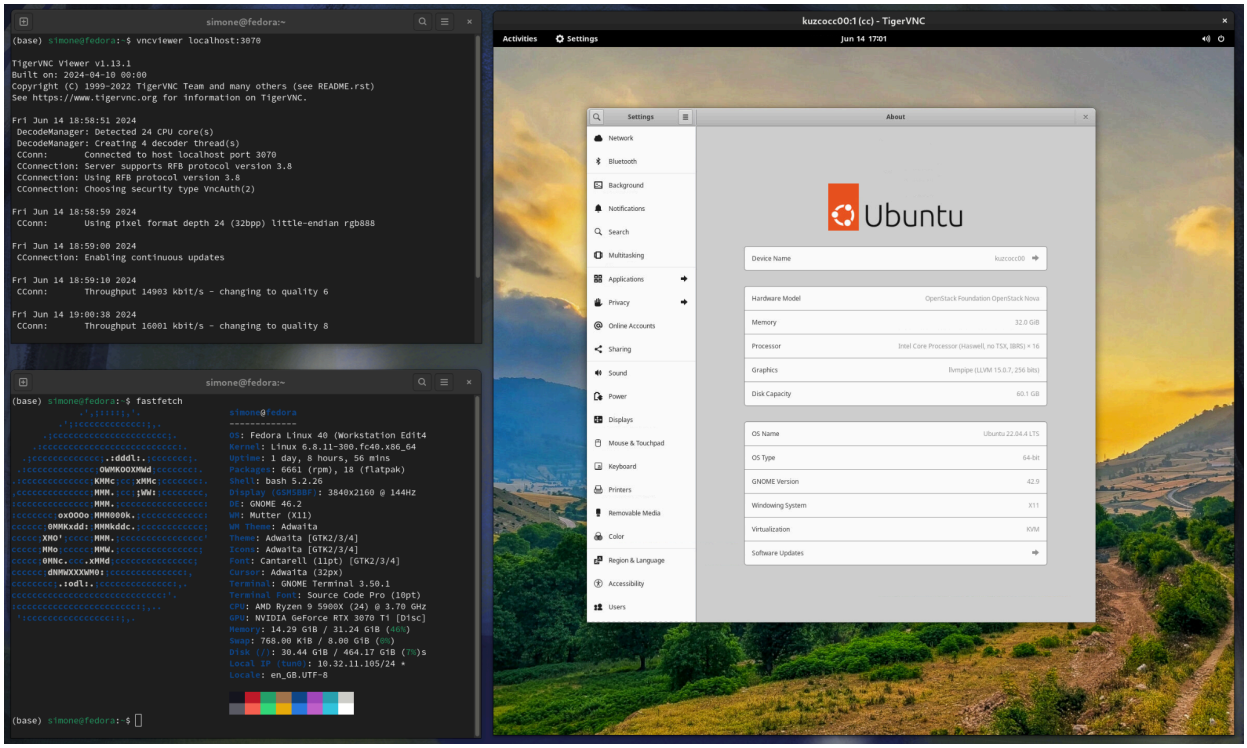
```
vncviewer localhost:local_port
```

Replace with the local port you chose before.

5. Conclusion

This guide has shown you how to install a desktop environment on an Ubuntu VM and configure TigerVNC for remote control using SSH. We explored different Ubuntu flavors (classical Ubuntu, Kubuntu, Xubuntu, Lubuntu) and specific steps for each. Additionally, we provided an introduction to the key concepts of desktop environments on Linux and their main components.

Here is an example of a good job done! :)



At the top left corner you can see a terminal that opens the connection with the server, at the bottom left corner, you can see a terminal showing my local machine and finally, at right side, the interface of Ubuntu with XFCE on my remote VM.

Remember to stop the VNC session and SSH tunneling when you're finished for security reasons with the following command:

```
vncserver -kill :1
```

6. Just some nuts and bolts

6.1. What is a Desktop Environment (DE)?

A Desktop Environment (DE) on Linux is a suite of software applications designed to provide a comprehensive Graphical User Interface (GUI) for users. It typically includes:

- **Window Manager**: Manages the placement and appearance of windows.
- **File Manager**: Allows users to navigate and manage files and directories.
- **Panels and Menus**: Provide access to applications, system settings, and other functionalities.
- **Graphical Toolkits**: Libraries like GTK+ for GNOME or Qt for KDE, used to draw the graphical user interface elements.

- **Utilities and Applications**: Basic programs such as terminal emulators, text editors, and system configuration tools.

A DE makes using the operating system more user-friendly by providing a consistent look and feel, along with integrated tools and utilities.

6.2. What is a Display Manager?

A Display Manager is a graphical login manager that starts a display server and manages user logins. It provides a login screen where users can enter their credentials to access the desktop environment. Key functions of a Display Manager include:

- **Starting the Display Server**: Initiates the X server (Xorg) or Wayland, which handles the graphical display.
- **User Authentication**: Provides a graphical interface for users to enter their username and password.
- **Session Management**: Allows users to choose which desktop environment or window manager to start.

Common display managers include:

- **GDM (GNOME Display Manager)**: Used by GNOME.
- **SDDM (Simple Desktop Display Manager)**: Used by KDE Plasma and LXQt.
- **LightDM**: Often used by Xubuntu and other lightweight DEs.

6.3. What is Xorg?

Xorg, also known as the X server or X Window System, is a crucial component of the graphical infrastructure on Unix-like operating systems, including Linux. It is responsible for the following:

- **Rendering Graphics**: Handles the drawing of windows, icons, and other graphical elements on the screen.
- **Input Handling**: Manages input devices like keyboards and mice, and passes input events to the appropriate applications.
- **Network Transparency**: Allows applications to run on one machine while displaying on another over a network.

6.3.1. Components of Xorg

1. **X Server**: The core component that interacts directly with the hardware (graphics card, monitor, input devices) and manages the display.
2. **X Client**: Applications that communicate with the X server to render their graphical interfaces.
3. **X Protocol**: The communication protocol between the X server and X clients.

4. **Xlib/XCB**: Libraries that provide an interface for applications to communicate with the X server using the X protocol.

6.3.2. How Xorg Works

1. **Starting the X Server**: When the system boots, the display manager starts the X server.
2. **Launching X Clients**: Applications, window managers, and desktop environments (X clients) connect to the X server to display their graphical interfaces.
3. **Handling Input and Output**: The X server manages input from devices like keyboards and mice and sends it to the appropriate X client. It also manages the output to the display.
4. **Network Transparency**: Xorg can run applications on one machine and display them on another, enabling remote graphical sessions.

Xorg serves as the backbone of the graphical user interface on many Linux distributions, providing the essential services needed to run and interact with graphical applications.

6.4. What is TigerVNC?

TigerVNC is a high-performance, platform-neutral implementation of VNC (Virtual Network Computing) that enables users to remotely control another computer over a network. TigerVNC extends the original VNC protocol and is optimized for enhanced performance and security.

6.4.1. Components of TigerVNC

1. **VNC Server:**
    - The server component runs on the machine you want to control remotely. It captures the display output and sends it to the VNC client.
2. **VNC Viewer (Client):**
    - The client component runs on the machine from which you want to control the remote system. It receives the display output from the VNC server and displays it to the user.
3. **Session Management:**
    - TigerVNC supports multiple sessions, allowing different users to connect and work on the same remote machine simultaneously.

6.4.2. Typical Use Cases

1. **Remote Technical Support:**
    - IT support staff can use TigerVNC to remotely troubleshoot and resolve issues on users' machines.
2. **Remote Work:**
    - Users can access and control their office computers from home or other remote locations.

3. **Education and Training:**
    - Instructors can use TigerVNC to demonstrate software and computing concepts to students remotely.
4. **System Administration:**
    - System administrators can manage servers and other infrastructure remotely, without needing physical access to the machines.