

How to install GUI on CentOS 7 Minimal and set Gnome Graphical Environment to automatically load on system boot

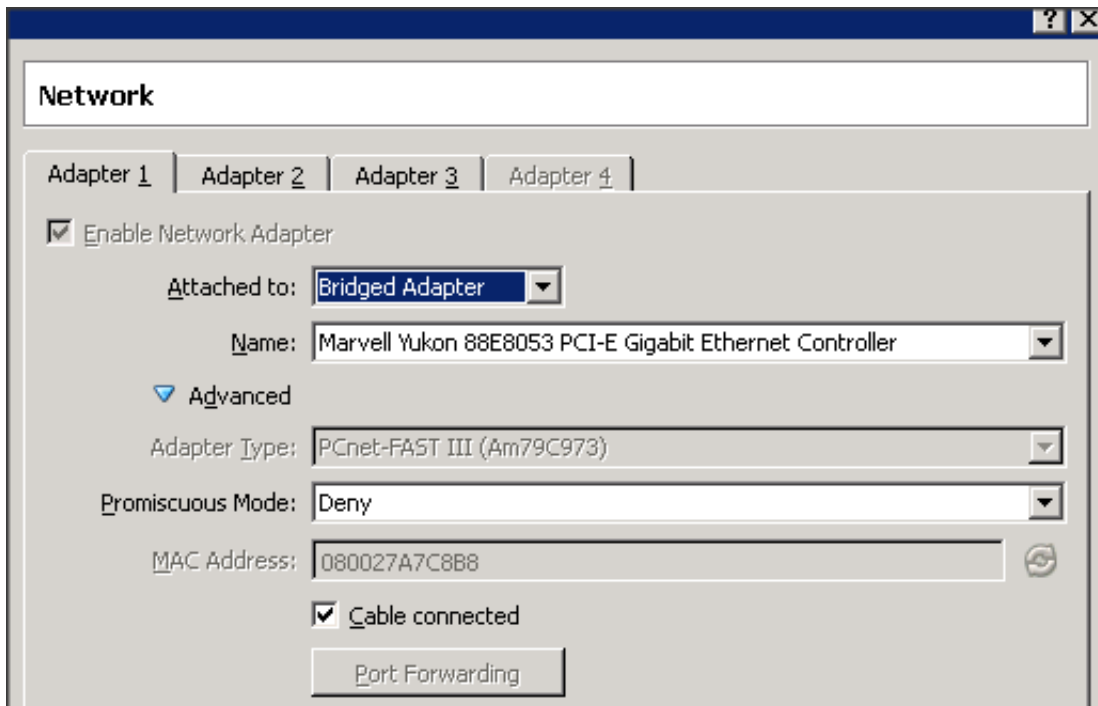
Author : admin



I have installed CentOS 7.7 Minimal Server Linux on a VirtualBox Virtual Environment as a test bed machine.

The system got installed easily successfully with the standard *CentOS python based graphical installer*, however I needed to place various software which was not there and for that of course I needed to have a network enabled.

To make network working instead of the default Network **NAT configuration** for the Virtual Machine I needed to use the *Network to be Attached to a Bridged Adapter* in order to make my *Windows machine* to provide network and (internet) access to VirtualMachine.



Then to make networking work after booting into CentOS I had to manually fetch IP via DHCP protocol with command:

```
[root@centos :~]# dhclient enp0s3
```

test@localhost:/home/test

File Edit View Search Terminal Help

```
[root@localhost test]# ifconfig -a
enp0s3: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 08:00:27:89:47:a0 txqueuelen 1000 (Ethernet)
    RX packets 3312 bytes 1443048 (1.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2941 bytes 219248 (214.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 66 bytes 5666 (5.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 66 bytes 5666 (5.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:29:c3:92 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0-nic: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 52:54:00:29:c3:92 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost test]# dhclient enp0s3
[root@localhost test]# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.136 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a387:5dad:a8f7:53de prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:89:47:a0 txqueuelen 1000 (Ethernet)
    RX packets 3321 bytes 1445053 (1.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2972 bytes 224087 (218.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost test]#
```

To make the setting permanent I had to also of course modify `/etc/sysconfig/network-scripts/ifcfg-enp0s3` file and change

ONBOOT=no

to

ONBOOT=yes

plugins) that enabled the CentOS Operating System to show in Virtualbox in fullscreen and to enable the Copy / Paste buffers to work from *The Hypervisor* (Windows in that case) and the Guest VM (the CentOS VM).

In CentOS terminology metapackages (a grouped package under a certain name, alias) are called simply groups) there is a "GNOME Desktop" group that can be used to install the GNOME Graphical Command from that point on with yum, like so:

```
[root@centos :~]# yum -y groups install "GNOME Desktop"
```

...

In a while the graphical environment will be in place, the command will install about 1300+ RPM packages, this will take about 5 minutes or so depending on your bandwidth connectivity. Once all is installed and configured successfully you can use the good old startx command to launch GNOME.

```
[root@centos :~]# startx
```



This of course will make Xserver and GNOME to run one time and on next reboot, you will end up in a plain text mode environment, so perhaps you will need to make the autolaunch of GNOME environment automatically on each boot in CentOS just like in most modern Linux distributions that use SYSTEMD to handle runlevels, you will need to configure it by changing the systemd default configured target via systemctl:

```
[root@centos :~]# systemctl list-units --type target | egrep "eme|res|gra|mul"
graphical.target    loaded active active Graphical Interface
multi-user.target   loaded active active Multi-User System
```

```
[root@centos :~]# systemctl set-default graphical.target
```

multi-user.target

```
[root@centos :~]# systemctl set-default  
graphical.target
```

```
[root@centos :~]# systemctl set-default graphical.target  
graphical.target
```

Next step was to enable the **Guest Additions** to do so I had to install in advance 2 RPM packages **kernel-headers** and **kernel-devel**

```
[root@centos :~]# yum install -y kernel-headers kernel-devel
```

...

Then I had to mount and run the **VboxLinuxAdditions.run** script to enable them, i.e.:

```
[root@centos :~]# mkdir /mnt/cdrom  
[root@centos :~]# mount /dev/cdrom /mnt/cdrom  
...  
[root@centos :~]# cd /mnt/cdrom/  
[root@centos :~]# sh VboxLinuxAdditions.run
```



```
test@localhost:/mnt/cdrom
File Edit View Search Terminal Help
[root@localhost Desktop]# cd /mnt/cdrom/
[root@localhost cdrom]# ls
AUTORUN.INF  runasroot.sh          VBoxSolarisAdditions.pkg
autorun.sh   TRANS.TBL             VBoxWindowsAdditions-amd64.exe
cert         VBoxDarwinAdditions.pkg  VBoxWindowsAdditions.exe
NT3x        VBoxDarwinAdditionsUninstall.tool  VBoxWindowsAdditions-x86.exe
OS2         VBoxLinuxAdditions.run
[root@localhost cdrom]# sh VBoxLinuxAdditions.run
Verifying archive integrity... All good.
Uncompressing VirtualBox 6.1.2 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Removing installed version 6.1.2 of VirtualBox Guest Additions...
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel
3.10.0-1127.13.1.el7.x86_64.

VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted
[root@localhost cdrom]#
```