VNC

The **Virtual Network Computing** (**VNC**) is a graphical desktop sharing system that uses the Remote Frame Buffer protocol (RFB) to remotely control another computer. It transmits the keyboard and mouse events from one computer to another, relaying the graphical screen updates back in the other direction, over a network.(http://en.wikipedia.org/wiki/Virtual_Network_Computing#cite_note-1)

The recommended clients are TightVNC or TigerVNC (free, open source, available for almost any platform).

Create VNC password

Local VNC password should be set before the first login. Do use a strong password.

```
[username@login2 ~]$ vncpasswd Password: Verify:
```

Start vncserver

To access VNC a local vncserver must be started first and also a tunnel using SSH port forwarding must be established. See below for the details on SSH tunnels. In this example we use port 61.

You can find ports which are already occupied. Here you can see that ports " /usr/bin/Xvnc :79" and " /usr/bin/Xvnc :60" are occupied.

```
[username@login2 ~]$ ps aux | grep Xvnc username    5971  0.0  0.0
201072 92564 ?        SN   Sep22   4:19 /usr/bin/Xvnc :79 -desktop
login2:79  (username)  -auth  /home/gre196/.Xauthority  -geometry
1024x768  -rfbwait  30000  -rfbauth  /home/username/.vnc/passwd
-rfbport 5979 -fp catalogue:/etc/X11/fontpath.d -pn username   10296 0.0 0.0
131772 21076 pts/29   SN   13:01   0:01 /usr/bin/Xvnc :60 -desktop
login2:61 (username) -auth /home/username/.Xauthority -geometry
1600x900 -depth 16 -rfbwait 30000 -rfbauth /home/jir13/.vnc/passwd
-rfbport 5960 -fp catalogue:/etc/X11/fontpath.d -pn .....
```

Choose free port e.g. 61 and start your VNC server:

‘ [username@login2 ~]$ vncserver :61 -geometry 1600x900 -depth 16

New ‘login2:1 (username)’ desktop is login2:1

Starting applications specified in /home/username/.vnc/xstartup Log file is /home/username/.vnc/login2:1.log ‘

Check if VNC server is started on the port (in this example 61):

‘ [username@login2 .vnc]$ vncserver -list

TigerVNC server sessions:

X DISPLAY # PROCESS ID :61 18437 ‘

Another command:

‘ [username@login2 .vnc]$ ps aux | grep Xvnc

username 10296 0.0 0.0 131772 21076 pts/29 SN 13:01 0:01 /usr/bin/Xvnc :61 -desktop login2:61 (username) -auth /home/jir13/.Xauthority -geometry 1600x900 -depth 16 -rfbwait 30000 -rfbauth /home/username/.vnc/passwd -rfbport 5961 -fp catalogue:/etc/X11/fontpath.d -pn ‘

To access the VNC server you have to create a tunnel between the login node using TCP **port 5961** and your machine using a free TCP port (for simplicity the very same, in this case).

The tunnel must point to the same login node where you launched the VNC server, eg. login2. If you use just cluster-name.it4i.cz, the tunnel might point to a different node due to DNS round robin.

Linux/Mac OS example of creating a tunnel

At your machine, create the tunnel:

```
local $ ssh -TN -f username@login2.cluster-name.it4i.cz -L 5961:localhost:5961
```

Issue the following command to check the tunnel is established (please note the PID 2022 in the last column, you'll need it for closing the tunnel):

```
local $ netstat -natp | grep 5961 (Not all processes could be identified, non-owned process info will not be shown, you would have to be root to see it all.) tcp 0 0 127.0.0.1:5961 0.0.0.0:* LISTEN 2022/ssh tcp6 0 0 ::1:5961 :::* LISTEN 2022/ssh
```

Or on Mac OS use this command:

```
local-mac $ lsof -n -i4TCP:5961 | grep LISTEN ssh 75890 sta545 7u IPv4 0xfb062b5c15a56a3b 0t0 TCP 127.0.0.1:5961 (LISTEN)
```

Connect with the VNC client:

```
local $ vncviewer 127.0.0.1:5961
```

In this example, we connect to VNC server on port 5961, via the ssh tunnel. The connection is encrypted and secured. The VNC server listening on port 5961 provides screen of 1600x900 pixels.

You have to destroy the SSH tunnel which is still running at the background after you finish the work. Use the following command (PID 2022 in this case, see the netstat command above):

```
kill 2022
```

Windows example of creating a tunnel

Use PuTTY to log in on cluster.

Start vncserver using command vncserver described above.

Search for the localhost and port number (in this case 127.0.0.1:5961).

```
[username@login2 .vnc]$ netstat -tanp | grep Xvnc (Not all processes
could be identified, non-owned process info  will not be shown,
you would have to be root to see it all.) tcp         0       0
127.0.0.1:5961          0.0.0.0:*             LISTEN    24031/Xvnc
```

On the PuTTY Configuration screen go to Connection->SSH->Tunnels to set up the tunnel.

Fill the Source port and Destination fields. **Do not forget to click the Add button**.

Run the VNC client of your choice, select VNC server 127.0.0.1, port 5961 and connect using VNC password.

Example of starting TigerVNC viewer

In this example, we connect to VNC server on port 5961, via the ssh tunnel, using TigerVNC viewer. The connection is encrypted and secured. The VNC server listening on port 5961 provides screen of 1600x900 pixels.

Example of starting TightVNC Viewer

Use your VNC password to log using TightVNC Viewer and start a Gnome Session on the login node.

Gnome session

You should see after the successful login.

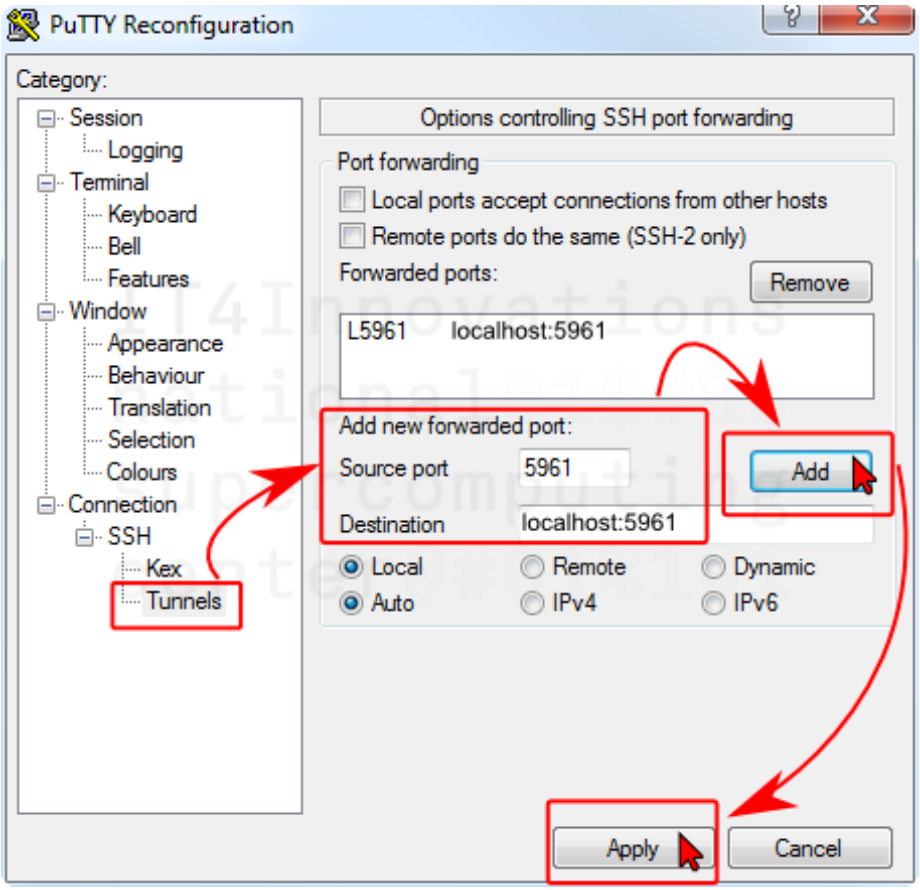

Figure 1:

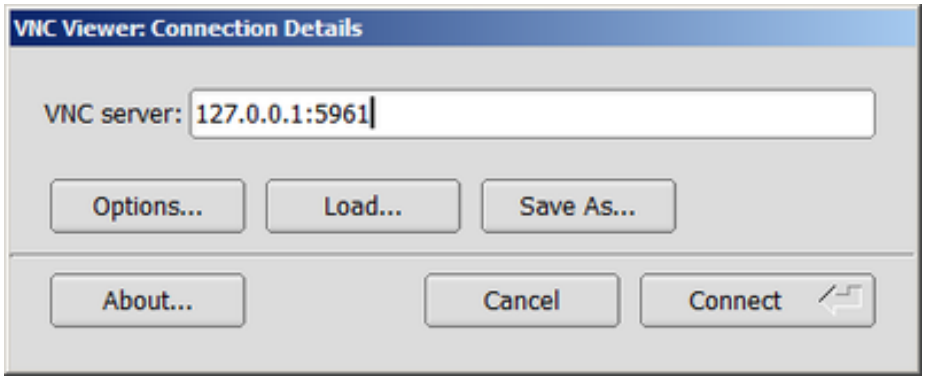

Figure 2:

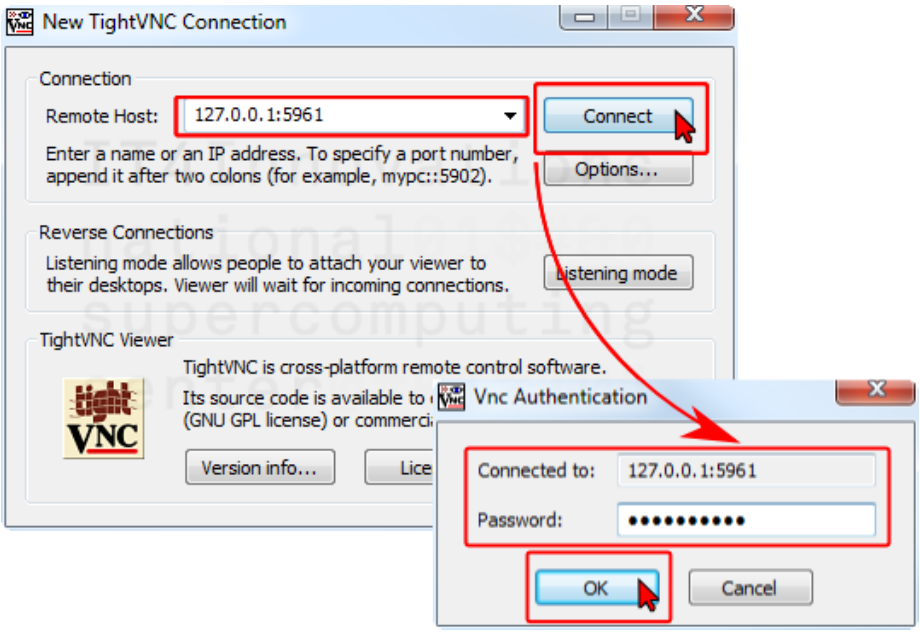

Figure 3:

Disable your Gnome session screensaver

Open Screensaver preferences dialog:

Uncheck both options below the slider:

Kill screensaver if locked screen

If the screen gets locked you have to kill the screensaver. Do not to forget to disable the screensaver then.

‘ [username@login2 .vnc]$ ps aux | grep screen username 1503 0.0 0.0 103244 892 pts/4 S+ 14:37 0:00 grep screen username 24316 0.0 0.0 270564 3528 ? Ss 14:12 0:00 gnome-screensaver

[username@login2 .vnc]$ kill 24316 ‘

Kill vncserver after finished work

You should kill your VNC server using command:

```
[username@login2 .vnc]$ vncserver -kill :61 Killing Xvnc process ID 7074 Xvnc process ID 7074 already killed
```

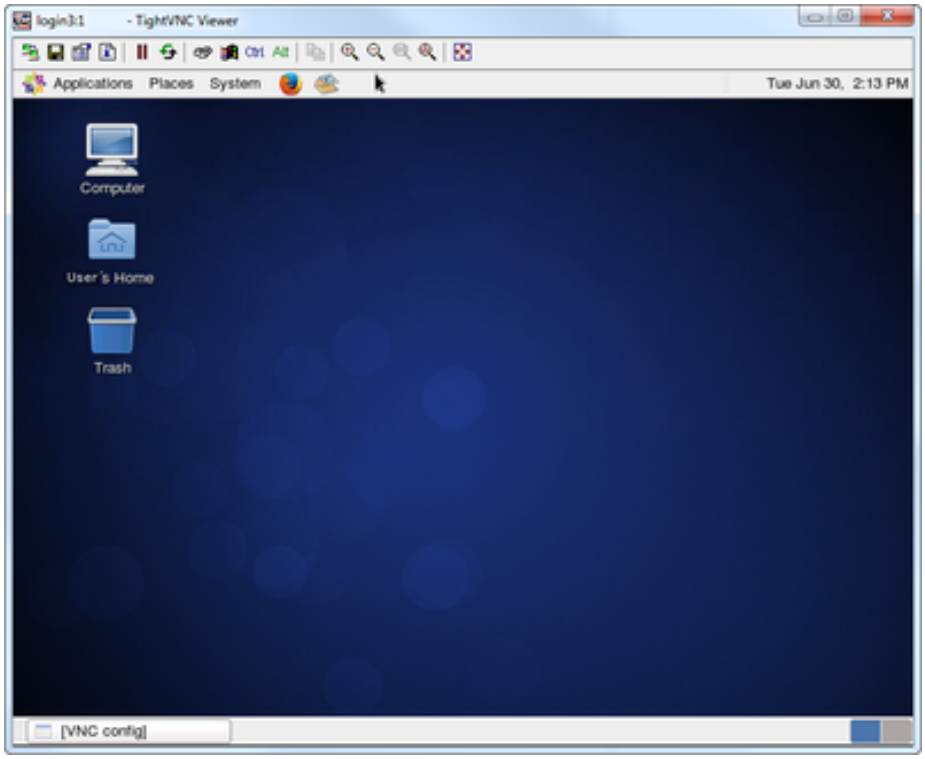
login3:1 - TightVNC Viewer
Ctrl
Alt
Applications
Places
System
Tue Jun 30, 2:13 PM
Computer
User's Home
Trash
[VNC config]

Figure 4:

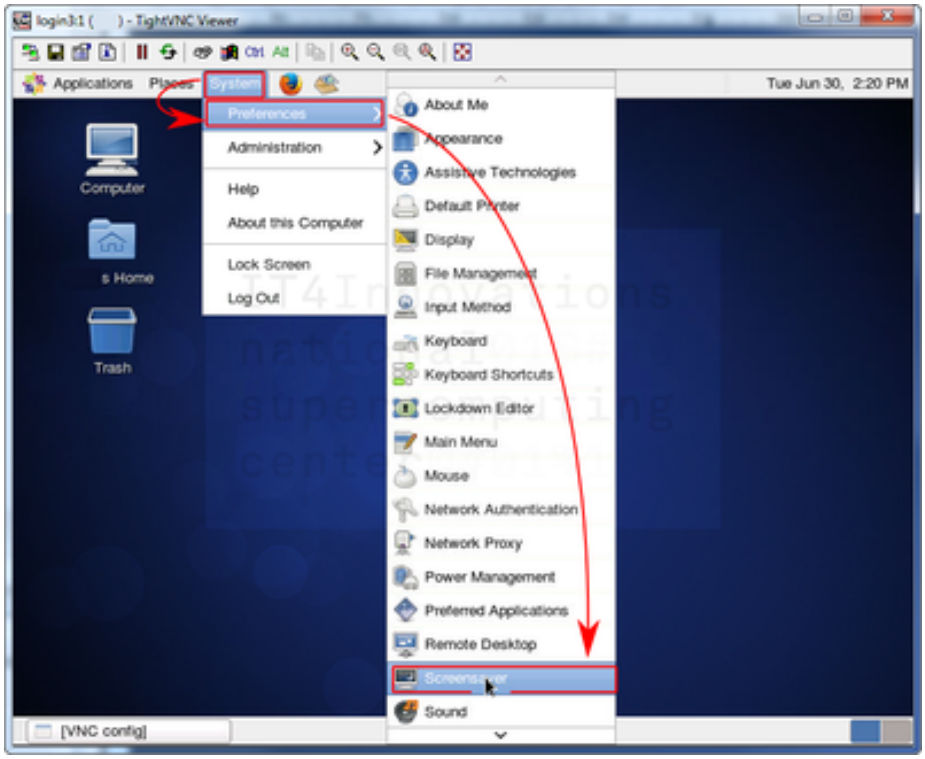
login3:1 ( ) - TightVNC Viewer
Applications
Places
System
Tue Jun 30, 2:20 PM
Preferences
Administration
Help
About this Computer
Lock Screen
Log Out
Computer
s Home
Trash
About Me
Appearance
Assistive Technologies
Default Printer
Display
File Management
Input Method
Keyboard
Keyboard Shortcuts
Lockdown Editor
Main Menu
Mouse
Network Authentication
Network Proxy
Power Management
Preferred Applications
Remote Desktop
Screensaver
Sound
[VNC config]

Figure 5:

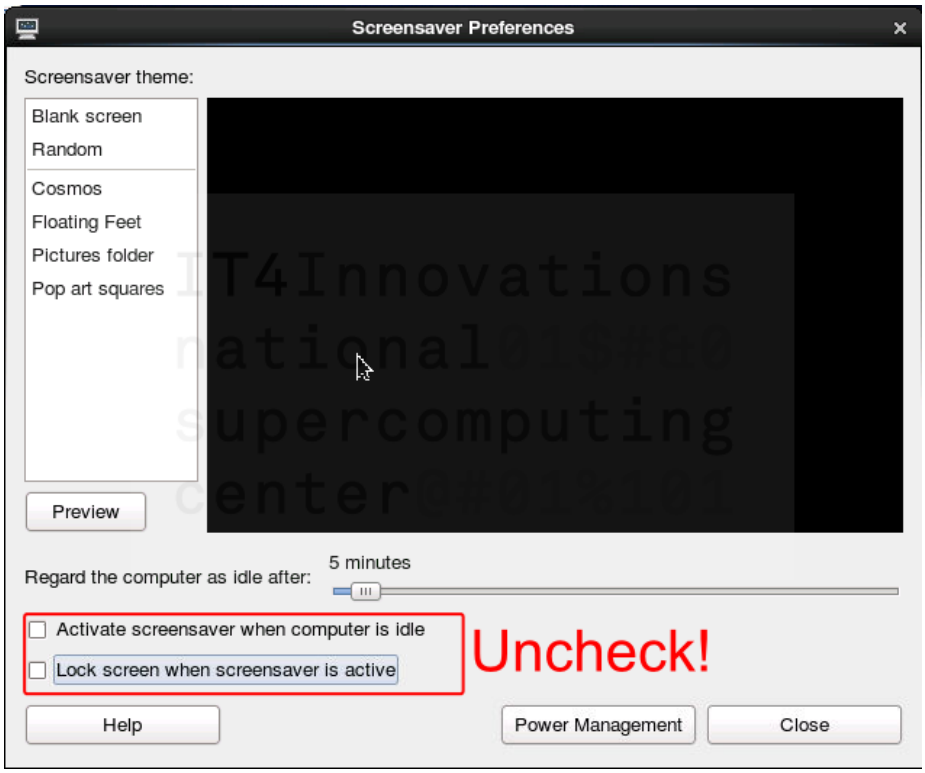
Screensaver Preferences
Screensaver theme:
Blank screen
Random
Cosmos
Floating Feet
Pictures folder
Pop art squares
Preview
Regard the computer as idle after:
5 minutes
Activate screensaver when computer is idle
Lock screen when screensaver is active
Uncheck!
Help
Power Management
Close

Figure 6:

Or this way:

```
[username@login2 .vnc]$  pkill vnc
```

GUI applications on compute nodes over VNC

The very same methods as described above, may be used to run the GUI applications on compute nodes. However, for maximum performance, proceed following these steps:

Open a Terminal (Applications -> System Tools -> Terminal). Run all the next commands in the terminal.

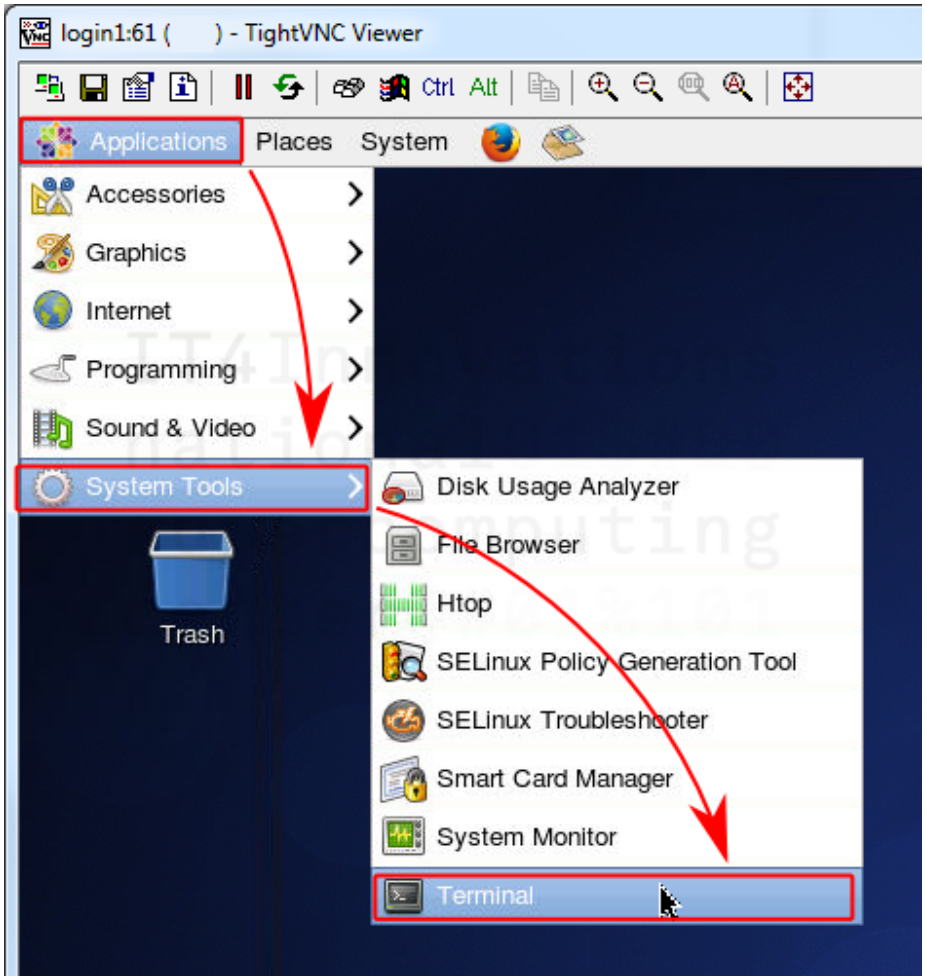

Figure 7:

Allow incoming X11 graphics from the compute nodes at the login node:

```
$ xhost +
```

Get an interactive session on a compute node (for more detailed info look here). Use the **-v DISPLAY** option to propagate the DISPLAY on the compute node. In this example, we want a complete node (24 cores in this example) from the production queue:

```
$ qsub -I -v DISPLAY=$(uname -n):$(echo $DISPLAY | cut -d ':' -f 2)
-A PROJECT_ID -q qprod -l select=1:ncpus=24
```

Test that the DISPLAY redirection into your VNC session works, by running a X11 application (e. g. XTerm) on the assigned compute node:

```
$ xterm
```

Example described above:

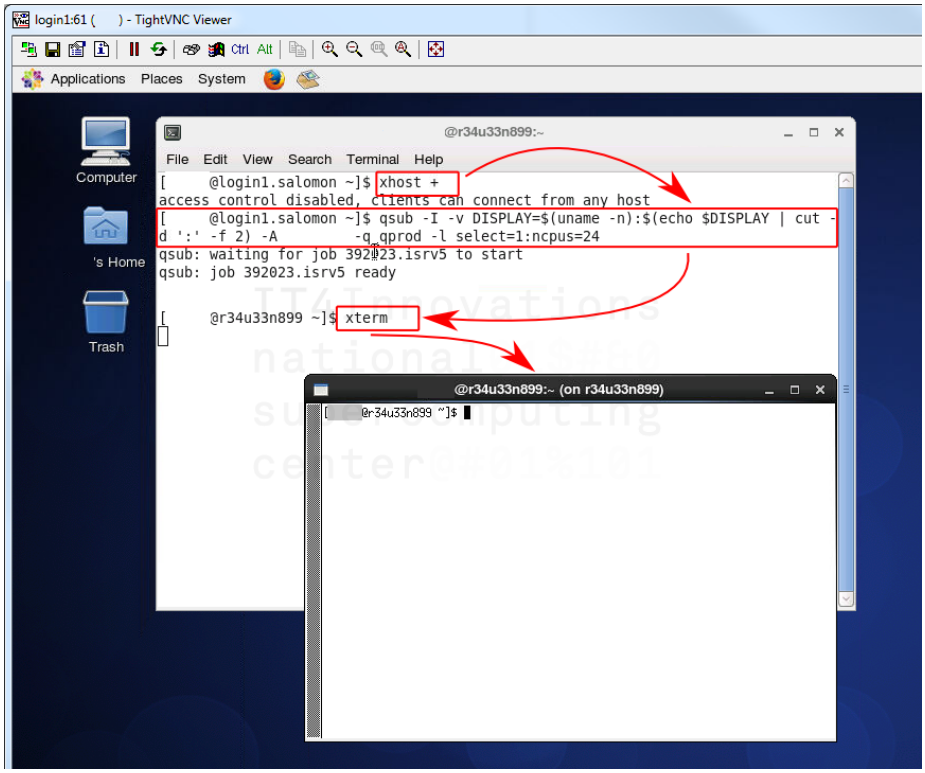

Figure 8: