

Utilisation des nœuds de visualisation

L'utilisation des nœuds de visualisation se fait par la réservation du nœud via un job slurm. Elle nécessite l'utilisation d'une session ssh avec une émulation X11 et d'un client VNC. Les étapes pour utiliser un nœud de visualisation sont les suivantes :

1. Connectez-vous sur muse-login.
2. Lancez le job qui va réserver le nœud (voir exemple de script visu.slurm).

```
[chapuis@muse-login01 ~]$ sbatch visu.slurm
```

Remarque : Vous trouverez un lien de télécharger du script visu.slurm à la fin du document. Si vous utilisez ce scripte, pensez à renseigner le champ #SBATCH --account= avec le nom de votre account dédié à la visu.
3. Slurm va créer en sortie un fichier visu.out contenant les paramètres de connexion au nœud de visu.

Remarque : le nom du fichier de sortie dépend de la configuration du fichier sbatch.

```
[chapuis@muse-login01 ~]$ cat visu.out
account:   admin
username:  chapuis
hostname:  visu01
```

```
vnc_host:  visu01:1
password:  hR9bixp6
To setup the tunnel, run the following command from your desktop:
ssh -L 5901:visu01:5901 chapuis@muse-login.meso.umontpellier.fr
After successful login you can access the VNC session at
localhost:5901 on your desktop
```

4. Recopiez la ligne ssh du fichier visu.out pour ouvrir une nouvelle session sur muse-login. Cette session permettra de créer un tunnel ssh vers le nœud de visu. La session ssh peut être ouverte depuis un terminal Linux, un terminal Xquartz sous MacOS ou un client ssh sous Windows. Je conseille aux utilisateurs de Windows d'utiliser le logiciel MobaXterm <https://mobaxterm.mobatek.net/download.html>. L'utilisation de Putty est possible mais nécessite une configuration plus complexe.
ATTENTION : Cette session ssh doit rester active durant toute la durée de l'utilisation du nœud de visu.

Remarque : Les captures d'écran ci-dessous sont faites sous MobaXterm et RealVNC viewer <https://www.realvnc.com/fr/connect/download/viewer/>.

```

• MobaXterm Personal Edition v20.5 •
(Outils Unix et serveur X pour Windows)

> Vos disques durs sont accessibles au travers du dossier /drives
> Le DISPLAY est positionné à 192.168.1.14:0.0
> Lors d'une connexion SSH, le DISPLAY est automatiquement exporté
> Le statut de chaque commande est indiqué par un symbole (✓ ou ✗)

• Important:
Vous utilisez la version personnelle de MobaXterm.
En achetant la version professionnelle de MobaXterm, vous pourrez
personnaliser le logiciel à votre convenance, en pré-définissant
vos options, votre propre message de bienvenue, votre logo ainsi
que plusieurs autres paramètres.
En plus de vous assurer un support professionnel, nous pouvons
modifier le programme MobaXterm ou développer de nouveaux plugins
selon vos besoins. Pour plus d'informations, utilisez Ctrl + Clic
sur le lien suivant : https://mobaxterm.mobatek.net/download.html

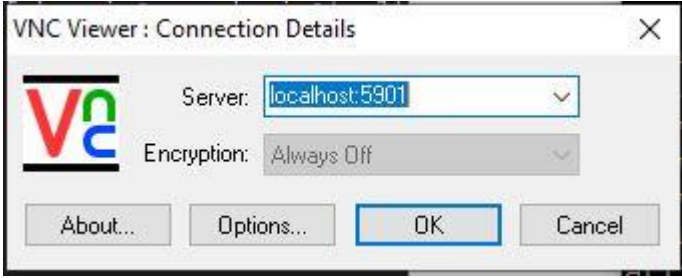
09/04/2021 11:20:57 /home/mobaxterm ssh -L 5901:visu01:5901 chapuis@muse-login.meso.umontpellier.fr
chapis@muse-login.meso.umontpellier.fr's password:
Last login: Fri Apr 9 11:29:44 2021 from lfbn-mon-1-976-230.w86-227.abo.wanadoo.fr

#####
#                               #
#      Bienvenue sur le cluster Muse      #
#                               #
#####

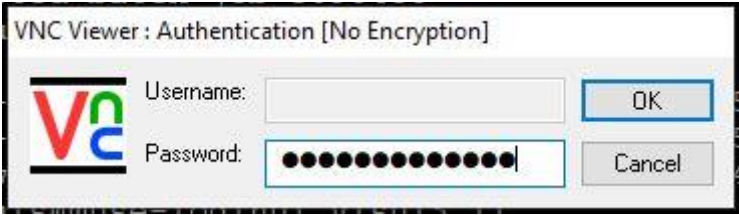
[chapis@muse-login01 ~]$

```

5. Une fois le tunnel ouvert, lancez un client VNC et connectez-vous à l'adresse indiquée dans le fichier de sortie visu.out.

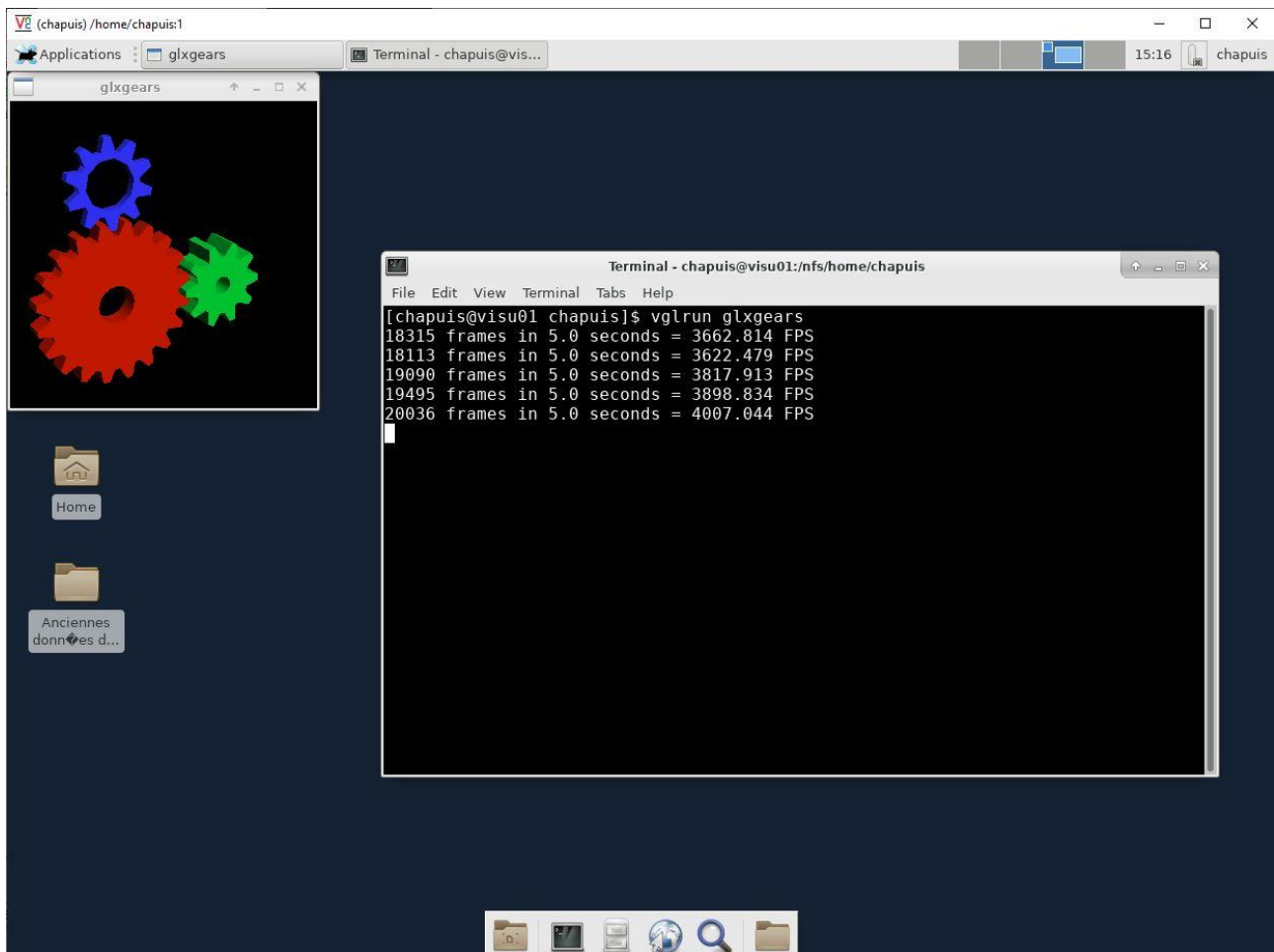


Le mot de passe à utiliser est celui indiqué dans le fichier de sortie visu.out.



- 6. Une fois la session VNC ouverte, il est possible de :
 - a. Ouvrir un terminal en ligne de commande
 - b. Lancer un job graphique utilisant les ressources graphiques de la carte GPU, à l'aide de la commande vglrun.

ATTENTION : L'utilisation de la commande vglrun est obligatoire pour utiliser les ressources de la carte GPU.



7. Une fois les travaux terminés il convient de stopper le job de réservation du nœud pour arrêter le compteur d'utilisation des ressources, à l'aide de la commande scancel :
[chapuis@muse-login01 ~]\$ scancel 3098438

Remarque : Si vous oubliez de stopper le job un walltime par défaut est configuré à 1 heure.

ATTENTION : Si vous souhaitez utiliser le nœud plus d'une heure il vous faut configurer l'option #SBATCH --time= dans le fichier visu.slurm.

Exemple de fichier visu.slurm

```
#!/bin/bash

#SBATCH --job-name=TurboVNC
#SBATCH --output=visu.out
#SBATCH --account=<votre_account_visu>
#SBATCH --partition=muse-visu
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1

# Define default desktop environment parameters
[[ -z $VNC_VIEWONLY ]] && export VNC_VIEWONLY="0"

# Define TurboVNC and scripts location
VGL_DIR="/opt/VirtualGL/bin"
TVNC_DIR="/opt/TurboVNC/bin"
LJPG_DIR="/opt/libjpeg-turbo/bin"

# Add TurboVNC/VirtualGL in PATH
for dir in ${TVNC_DIR} ${VGL_DIR} ${LJPG_DIR} ${SVIS_DIR}/bin; do
    [[ ":${PATH}:" != *"${dir}:"* ]] && export PATH=${dir}:${PATH}
done

# Collect user information
VISU_USER=${SLURM_JOB_USER}
VISU_HOME=$(eval echo ~${VISU_USER})
VISU_UID=$(id -u ${VISU_USER})
# Define commands and scripts
VNC_SERVER="${TVNC_DIR}/vncserver"
XSTARTUP="${VISU_HOME}/.vnc/xstartup.xfce4"

# Write custom xstartup
if [[ ! -d $(dirname ${XSTARTUP}) ]]; then mkdir -p $(dirname ${XSTARTUP}); fi
cat > ${XSTARTUP} << EOF
#!/bin/sh

# Uncomment the following two lines for normal desktop:
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
export XDG_SESSION_TYPE=x11

# Define TurboVNC and scripts location
VGL_DIR="/opt/VirtualGL/bin"
TVNC_DIR="/opt/TurboVNC/bin"
LJPG_DIR="/opt/libjpeg-turbo/bin"

# Add TurboVNC/VirtualGL in PATH
for dir in ${TVNC_DIR} ${VGL_DIR} ${LJPG_DIR}; do
    [[ ":${PATH}:" != *"${dir}:"* ]] && export PATH=${dir}:${PATH}
done

# Start some apps if you like
#xterm -e "vglrun glxgears" &
#xterm -e "nvidia-smi -l 1" &
#xclock &

# Start desktop environment
TVNC_VM=/usr/bin/startxfce4
exec \${TVNC_VGLRUN} \${TVNC_VM}

EOF
chmod +x ${XSTARTUP}

# Fix for dconf
export XDG_RUNTIME_DIR=/tmp/${VISU_UID}
mkdir -p ${XDG_RUNTIME_DIR}

# Generate VNC password
VISU_PASS=$(openssl rand -base64 6)
if [[ ${VNC_VIEWONLY} -eq 1 ]];then
    VISU_VOPASS=$(openssl rand -base64 6)
fi

# Set VNC Password
[[ -d ${VISU_HOME}/.vnc ]] || mkdir -p ${VISU_HOME}/.vnc
if [[ ${VNC_VIEWONLY} -eq 0 ]];then
```

```

        #echo "${VISU_PASS}"|vncpasswd -f > ${VISU_HOME}/.vnc/passwd
        /usr/bin/expect <<EOF > /dev/null
spawn ${TVNC_DIR}/vncpasswd
expect "Password:"
send "${VISU_PASS}\r"
expect "Verify:"
send "${VISU_PASS}\r"
expect "Would you like to enter a view-only password (y/n)?"
send "n\r"
expect eof
exit
EOF
else
    /usr/bin/expect <<EOF > /dev/null
spawn ${TVNC_DIR}/vncpasswd
expect "Password:"
send "${VISU_PASS}\r"
expect "Verify:"
send "${VISU_PASS}\r"
expect "Would you like to enter a view-only password (y/n)?"
send "y\r"
expect "Password:"
send "${VISU_VOPASS}\r"
expect "Verify:"
send "${VISU_VOPASS}\r"
expect eof
exit
EOF
fi

# Start VNC Server
MAX_VID=9
VISU_ID=1
VISU_HOST=$(hostname -s)
while test ${VISU_ID} -le ${MAX_VID}; do
    VNC_NAME="(${VISU_USER}) ${VISU_HOME}:${VISU_ID}"
    (${VNC_SERVER} -nohttpd -xstartup ${XSTARTUP} :${VISU_ID} -name "${VNC_NAME}")
2>&1) >/dev/null
    HRET=$?
    if test ${HRET} -eq 0; then break; fi
    ((VISU_ID++))
done

# Early exit
if [[ -z ${VNC_SERVER} -list 2>/dev/null|grep ^:${VISU_ID} ]]; then
    echo "Can't create a VNC server. Exit with error."
    exit 255
fi

# Collect VNC information
VISU_PORT=$((5900+VISU_ID))

# Print information
echo "account:  ${SLURM_JOB_ACCOUNT}"
echo "username:  ${VISU_USER}"
echo "hostname:  ${VISU_HOST}"
echo
echo "vnc_host:  ${VISU_HOST}:${VISU_ID}"
echo "password:  ${VISU_PASS}"
# Print information for view-only
if [[ ${VNC_VIEWONLY} -eq 1 ]];then
    echo
    echo "view-only:  ${VISU_VOPASS}"
fi

echo "To setup the tunnel, run the following command from your desktop:"
echo "ssh -L 5901:${VISU_HOST}:${VISU_PORT} $VISU_USER@muse-login.meso.umontpellier.fr"
echo "After successful login you can access the VNC session at localhost:5901 on your
desktop"

# Wait until end of the job to kill vncserver
endtime=$(scontrol show job ${SLURM_JOB_ID}|tr ' ' '\n'|grep EndTime|cut -d= -f2)
endtime=$(date -d "${endtime}" +%s)
curtime=$(date -d '+30 secs' +%s)
sleep $((endtime - curtime))

# Call cleaning
/etc/slurm/slurm.epilog.clean

```