
Documentation CaSciModOT

Version 1

CaSciModOT

octobre 18, 2022

1	Créer des clés SSH	1
1.1	Qu'est-ce qu'une clé ssh	1
1.2	Vous utilisez un ordinateur fonctionnant sous Linux ou MacOS	1
1.3	Vous utilisez un ordinateur fonctionnant sous Windows	2
2	Demander une ouverture de compte	4
3	Ouvrir un canal VPN	5
3.1	Introduction	5
3.2	Orléans	5
3.3	Tours	10
3.4	INSA	10
3.5	INRAE Nogent/Vernisson	10
3.6	CNRS Campus d'Orléans	11
4	Se connecter au cluster leto	12
4.1	Depuis Linux & MacOS	12
4.2	Depuis Windows	12
5	Gérer les fichiers	15
5.1	Transfert de fichiers de sa machine locale vers leto	15
5.2	Transfert de fichiers de leto vers sa machine locale	16
6	Soumettre une tâche	17
6.1	Soumettre une tâche en batch	17
6.2	Soumettre une tâche en interactif	18
7	Utiliser les modules	19
8	Utiliser les logiciels	21
8.1	Utiliser R	21
8.2	Utiliser Python	21

Créer des clés SSH

La connexion sécurisée au serveur d'accueil du cluster CaSciModOT nécessite un échange de clé cryptographique. Si vous ne disposez pas déjà d'une telle clé, il vous faudra la générer. Voici la démarche :

1.1 Qu'est-ce qu'une clé ssh

L'authentification par clé est plus forte que l'authentification par mot de passe.

Elle se base sur :

- Une clé publique : que vous fournirez aux serveurs où vous souhaitez vous connecter
- une clé privée : elle permet de prouver votre identité aux serveurs sur lesquels se trouve votre clé publique.

Avertissement : Cette clé ne doit pas être communiquée !

- Une passphrase : pour sécuriser votre clé privée

Il existe plusieurs types de chiffrement pour vos clés. Pour l'accès au cluster CaSciModOT, seules les **clés ed25519** sont acceptées.

1.2 Vous utilisez un ordinateur fonctionnant sous Linux ou MacOS

1. Vérifiez si vous possédez déjà une clef publique.

Elle se trouve dans le fichier `$HOME/.ssh/id_ed25519.pub`. Dans un terminal ouvert dans votre dossier HOME, tapez :

```
$ ls .ssh/id_ed25519.pub
```

Si vous obtenez en retour

```
.ssh/id_ed25519.pub
```

vous possédez une clef. Passez à l'étape [Demander une ouverture de compte](#)
Si vous obtenez :

```
ls: cannot access '.ssh/id_ed25519.pub': No such file or directory
```

vous ne possédez pas de clef, il vous faut en générer une en suivant les instructions à l'étape **2**.

- 2. Générez une clef publique.** Dans un terminal tapez la commande :

```
$ ssh-keygen -t ed25519
```

et acceptez les réponses par défaut en tapant la touche Entrée à chaque question. (Vous pouvez protéger votre clef par un mot de passe si vous le souhaitez. Dans ce cas, il vous faudra entrer ce mot de passe à chaque connexion à la grappe de calcul.) Vérifiez que vous avez bien généré une clef publique en vous référant à l'étape **1**.

Note : ssh-keygen génère un couple de fichiers `.ssh/id_ed25519.pub` (votre clé publique, à envoyer lors de la demande d'ouverture de compte) et `.ssh/id_ed25519` (votre clé privée qui sera automatiquement utilisée lors de vos connexions).

- 3. Demandez une ouverture de compte**

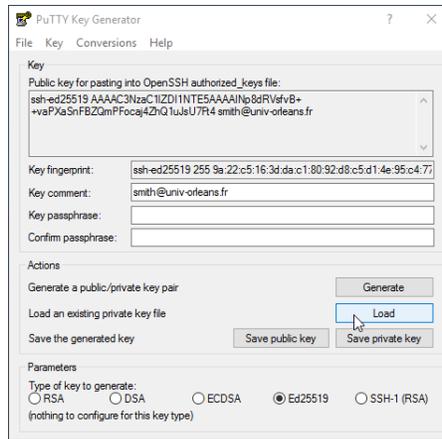
Une fois le compte créé, vous pouvez vous connecter à la grappe de calcul en suivant la procédure définie dans la page de [connexion au cluster leto](#)

1.3 Vous utilisez un ordinateur fonctionnant sous Windows

- 1. Télécharger et installez l'utilitaire PUTTY** depuis le lien :

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Installez l'utilitaire en laissant les options par défaut. Exécutez le programme puttygen



- Cochez Ed25519
- Cliquez sur **Generate** pour générer la clef
- Générer votre clé privée : pour cela :
 - **Key comment** : commentaire ex. votre adresse mail
 - **Key passphrase** : laissez vide
 - Cliquez sur **Save private Key** pour **sauvegarder la clef privée dans un fichier texte** (avec l'extension .pub) à envoyer lors de la demande d'ouverture de compte

Avertissement : Veillez à **ne pas** sauvegarder vos clés dans un répertoire temporaire. Ce répertoire sera consulté à chaque connexion avec putty. Vous pouvez créer un dossier C:\Users\nom_session_windows\.ssh dans lequel placer vos clés publique et privée.

2. Demandez une ouverture de compte

Une fois le compte créé, vous pouvez vous connecter à la grappe de calcul en suivant la procédure définie dans la page de [connexion au cluster leto](#)

CHAPITRE 2

Demander une ouverture de compte

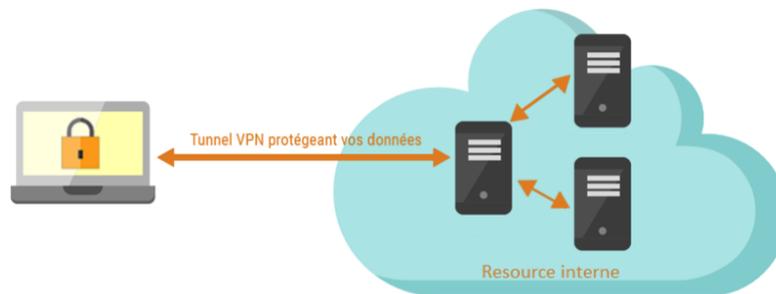
Demandez une ouverture de compte. Rendez-vous sur le site CaSciModOT : <https://test.cascimodot.fr/fr/node/8>

Renseigner le questionnaire. Il vous sera demandé de télécharger le fichier contenant votre clé publique (*.pub) que vous venez de créer.

Note : Sous **Linux & MacOS** la clé publique id_ed25519.pub se trouve dans le répertoire **~/.ssh**

3.1 Introduction

Le service d'accès distant vise la mise à disposition sécurisée de ressource interne de l'établissement (Université, laboratoire, etc.) depuis un poste maitrisé connecté à Internet.



Les accès à la machine Leto, maintenant hébergée au DataCentre-CVL, se fait essentiellement par VPN. Les modalités d'accès dépendent de votre établissement d'origine :

3.2 Orléans

3.2.1 Condition d'utilisation & prérequis

Utilisation de cet accès distant est destiné aux **personnels utilisant un poste professionnel** fourni par l'université d'Orléans.

Cette documentation s'adresse aux personnels ayant installé le client GlobalProtect téléchargé depuis les liens suivants :

- **Pour Windows & Mac** : <https://rpv.univ-orleans.fr>



Note : utilisez p<siham/harpege> pour *usrename*

— **Pour Linux :**

- <http://intranet.univ-orleans.fr/tic/cri/reseau/acces-au-reseau-du-campus-par-vpn/agent-globalprotect-pour-linux> ou
- https://www.univ-orleans.fr/siteD8/default/files/outils_numeriques/PanGPLinux.tgz

L'utilisation d'une version professionnelle de **Windows 10**, **MacOs** ou **Linux** et de l'antivirus **Kaspersky à jour** sont des prérequis au bon fonctionnement de la solution.

La connexion nécessite un compte (identifiant **pXXXX**) dans le domaine Windows global (Active Directory) de l'université **"CAMPUS"**. Si vous ne connaissez pas le mot passe de ce compte vous pouvez le définir à l'aide du portail <https://mdp.ad.univ-orleans.fr/>

Note :

1. s'authentifier d'abord avec son compte ENT (oXXXXX ou pXXXX)
2. Un formulaire demande à définir un nouveau mot de passe

Les ressources accessibles au travers de cet accès sont susceptibles d'évoluer dans les semaines à venir au fur et à mesure des collaborations avec votre équipe informatique locale.

3.2.2 Accéder au client GlobalProtect

— Windows

1. Dépliez le menu en bas à droite de votre écran.
2. Cliquez sur l'icône « GlobalProtect »



— Linux

Le client GlobalProtect se lance automatiquement à l'ouverture d'une session Linux, mais il peut aussi être démarré manuellement en ligne de commande :

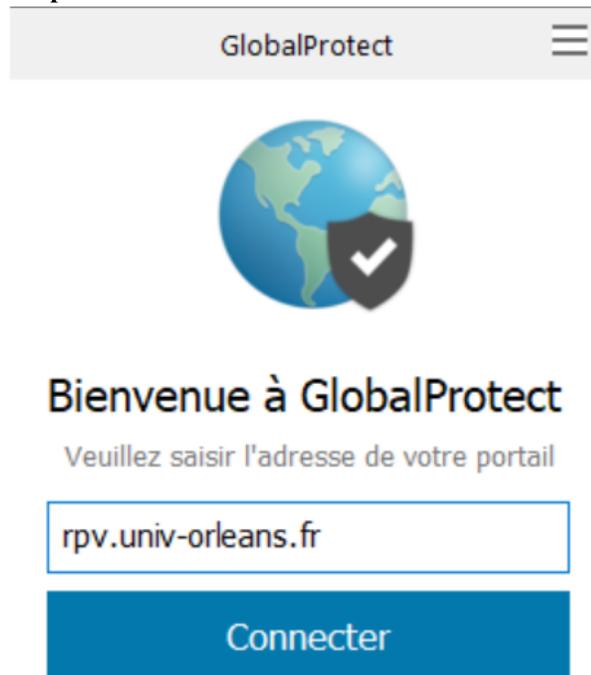
```
$ globalprotect lanch-ui
```

— MacOS

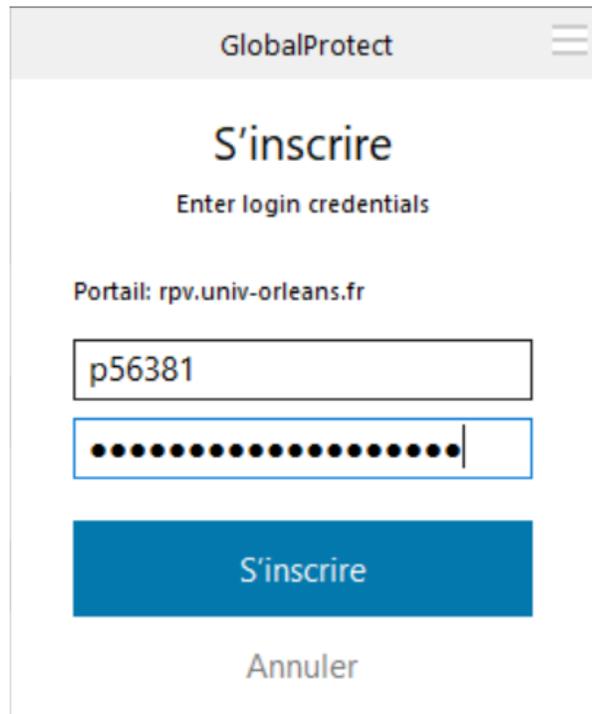
Cliquez sur l'icône « GlobalProtect » en haut à droite de votre écran

3.2.3 Paramétrer le client GlobalProtect (Windows/MacOS/Linux)

- Saisissez l'adresse du portail **rpv.univ-orleans.fr**



- Utilisez les identifiants de votre compte CAMPUS



GlobalProtect

S'inscrire

Enter login credentials

Portail: rpv.univ-orleans.fr

S'inscrire

Annuler

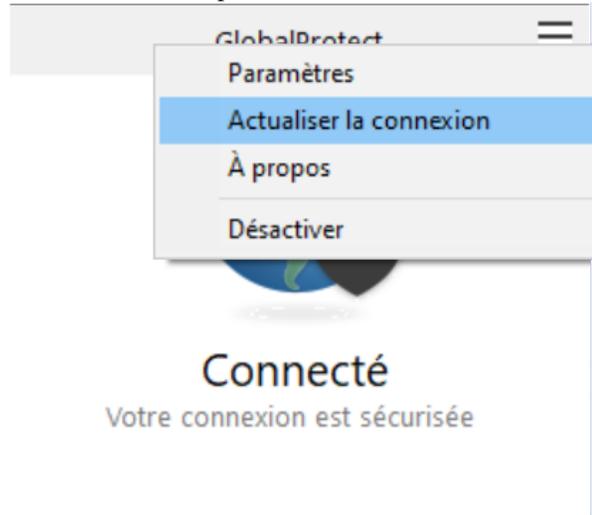
3.2.4 Contrôler le client GlobalProtect

- Une fois votre authentification validée vous devez observer l'un des deux écrans ci-dessous.
- Le client vous indique si vous êtes connecté depuis le réseau informatique interne de l'université ou depuis l'extérieur.

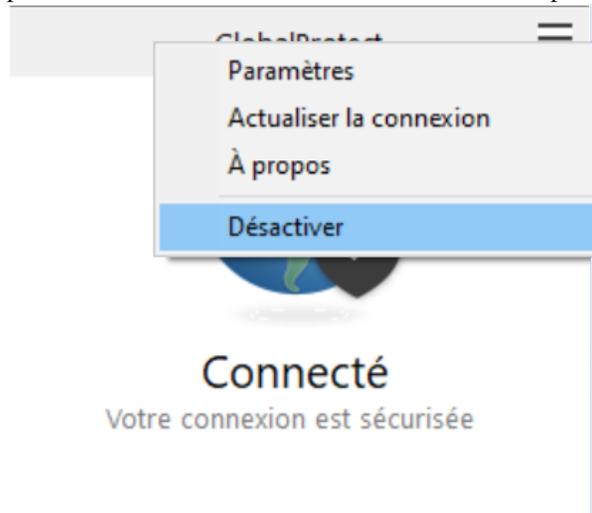


3.2.5 Piloter le client GlobalProtect

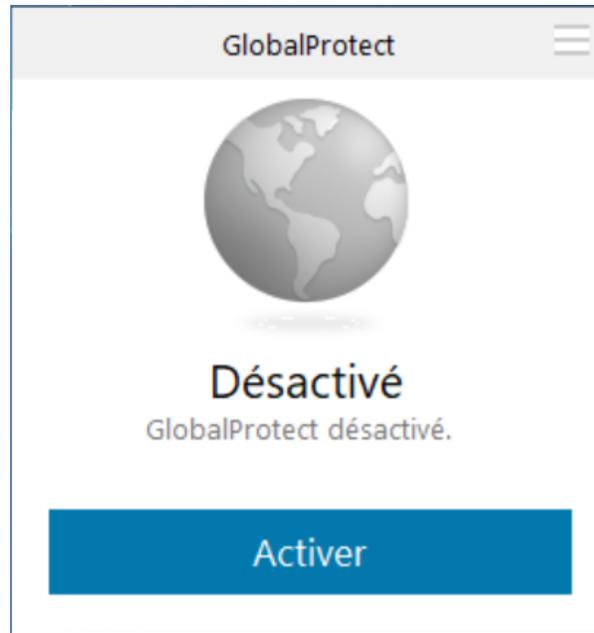
- Lors d'un changement de votre localisation, il peut être nécessaire de «Actualiser » votre connexion.



- En cas de difficultés, vous pouvez « Désactiver » le client et utiliser vos accès précédents.



- Nous vous recommandons de laisser le mode connexion automatique. Si vous avez désactivé le client, vous pouvez le relancer en cliquant sur son icône sous Windows, ou en exécutant la commande `globalprotect launch-ui` sous Linux.



3.3 Tours

- Vous opérez depuis votre poste professionnel connecté au **réseau de l'université de Tours** (filaire ou Wifi Eduroam), vous accédez **directement** à la ressource CaSciModOT via ssh.
- Vous opérez depuis votre poste professionnel depuis un **autre endroit** (domicile, autre..), vous devez utiliser la configuration **VPN** qui est installée sur votre poste pour accéder au service CaSciModOT.

Avertissement : Attention, pour que le service soit disponible, vous devez auparavant avoir demandé sur notre outil de gestion des demandes (<http://pal.univ-tours.fr>) rubrique « demander un droit d'accès » votre inscription dans le groupe des usagers CaSciModOT ; l'opération effectuée, votre VPN sera opérant et vous permettra de vous connecter.

3.4 INSA

Veillez vous adresser à M. Sylvain Lesage

3.5 INRAE Nogent/Vernisson

Veillez vous adresser à M. Philippe Guillemard

3.6 CNRS Campus d'Orléans

- de votre **laboratoire par ssh**
- de l'**extérieur par le VPN Pulse Secure**, voir votre informaticien

Avertissement : Ce n'est plus OpenVPN qui servait pour Artemis

Une fois la connexion VPN établie, vous pouvez passer à la [connexion au cluster](#)

Se connecter au cluster leto

4.1 Depuis Linux & MacOS

Connectez-vous à la grappe de calcul. Une fois votre compte effectivement ouvert par l'équipe d'administration, vous pourrez vous connecter au serveur d'accueil via SSH avec la commande suivante

```
$ ssh tdupont@leto.cascimodot.datacentre-valdeloire.fr
```

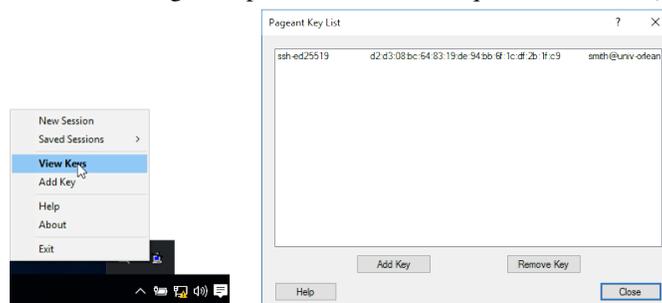
Cette connexion devra se faire depuis un ordinateur au sein de votre laboratoire. Pour vous connecter depuis un autre endroit (par exemple de chez vous), référez-vous à la page [Ouvrir un canal VPN](#)

4.2 Depuis Windows

1. Lancez l'agent **Putty** en exécutant la commande **pageant**

Cliquez sur menu contextuel de pageant (clic droit) et affichez les clef privées avec **View Keys**

Ajoutez la clef privée créée et sauvegardée précédemment en cliquant sur **Add Key**



Avertissement : pageant doit être exécuté à chaque nouvelle session Windows

2. **Connectez-vous** à la grappe de calcul avec Putty en exécutant depuis une invite de commandes :

```
> putty votre_login@leto.cascimodot.datacentre-valdeloire.fr
```

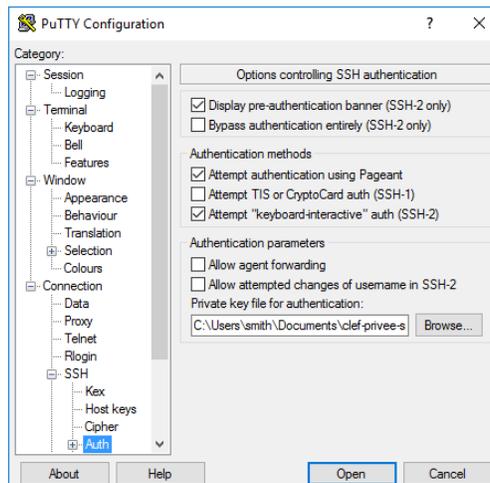
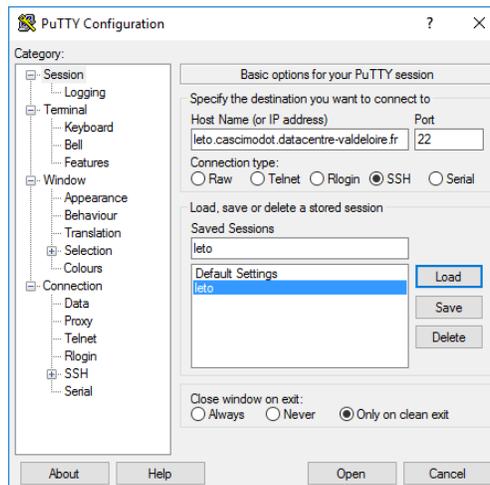
3. **Copiez vos fichiers** vers/depuis leto en exécutant le programme *psftp* :

Après le prompt psftp> taper la commande suivante :

```
psftp> open votre_login@leto.cascimodot.datacentre-valdeloire.fr
```

4. **Sauvegardez votre session SSH.**

Pour ne pas avoir à renseigner votre clé privée à chaque connexion dans pageant vous pouvez sauvegarder votre session putty : Une fois putty lancé



- **Connection>Data**
Auto login username : votre nom d'utilisateur sur leto
- **Connection>SSH>Auth**

Private keyfile for authentication : cliquez sur **Browse...** et sélectionnez le fichier contenant votre clé privée.

— **Session>**

- Host Name : leto.cascimodot.datacentre-valdeloire.fr
- Saved Sessions : donner un nom, exemple *leto*, et cliquer sur le bouton **Save**

Note : Pour vous connecter depuis l'extérieur de votre laboratoire. Ceci nécessite de passer par un [tunnel VPN](#), qui vous est fourni par votre laboratoire ou votre organisme de rattachement (Université, BRGM, CNRS, etc.). Rapprochez-vous de votre responsable informatique. Une fois le tunnel VPN établi, vous pourrez vous connecter normalement à la grappe de calcul en suivant les instructions **1** à **4**.

5.1 Transfert de fichiers de sa machine locale vers leto

Le téléchargement de fichier vers/depuis le frontal se fait avec la commande `scp`.

Exemples de transfert depuis le répertoire courant de votre machine vers Leto.

- Copie du fichier `file` dans le répertoire `fichier` (déjà créé sur Leto) :

```
scp file cmaltese@leto.cascimodot.datacentre-valdeloire.fr:fichier
```

- Copie des fichiers `file-a`, `file-b`, `file-c` dans le répertoire `fichier` :

```
scp file-? cmaltese@leto.cascimodot.datacentre-valdeloire.fr:fichier
```

le `?` remplace 1 caractère.

- Copie du répertoire `data` dans le répertoire `fichier`

```
scp -r data cmaltese@leto.cascimodot.datacentre-valdeloire.fr:fichier
```

l'option `-r` (pour récursif) transfère le contenu du répertoire `data` (y compris les répertoires appartenant à `data`)

5.2 Transfert de fichiers de leto vers sa machine locale

Exemples de transfert de la machine leto vers le répertoire courant de votre machine.

- Copie du fichier file du répertoire fichier (déjà créé sur leto) en local :

```
scp cmaltese@leto.cascimodot.datacentre-valdeloire.fr:fichier/file .
```

Le point . (en toute fin de ligne) représente le répertoire courant sur votre machine

- Copie des fichiers file-a.dat, file-b.tex, file-c.gp du répertoire fichier en local

```
scp cmaltese@leto.cascimodot.datacentre-valdeloire.fr:fichier/file-*. * .
```

le * remplace n'importe quelle chaîne de caractères.

Depuis une machine Windows, vous pouvez utiliser l'utilitaire WinSCP.

Soumettre une tâche

Les jobs se lancent soit en différé (batch), soit en interactif. Dans les 2 cas il faut réserver des ressources sur une file d'attente. En aucun cas, on ne lance un job directement, car alors il tourne sur les machines frontales et non sur les machines de calcul. Les files d'attente ont été créées pour avoir des ensembles de machines homogènes :

- cpu
- gpu-intel
- gpu-amd

Vous pouvez trouver la liste complète des commandes SLURM à l'adresse <https://slurm.schedmd.com/pdfs/summary.pdf>

Par défaut la durée d'un job est de 12h.

6.1 Soumettre une tâche en batch

```
sbatch script.job
```

où `script.job` est le fichier qui contient la réservation des ressources et le lancement de votre job. À sa création, votre compte comporte le répertoire `example` qui contient 3 exemples de script de soumission :

- `generic.job`
- `parallel_on_home.job`
- `parallel_on_scratch.job`

La différence essentielle concerne le disque de stockage de vos données lors de l'exécution de votre job. Vous pouvez soit travailler localement sur le scratch de chaque noeud (exemple `parallel_on_scratch.job`) soit dans votre home (exemple `parallel_on_home.job`).

Avertissement : Contrairement à artemis il n'y a pas de scratch global visiblement de tous les noeuds. Ici chaque noeud a son propre espace scratch.

6.2 Soumettre une tâche en interactif

Les instructions suivantes permettent de connaître l'état de la machine (ce qui est utile pour connaître les ressources disponibles), d'allouer des ressources, de lancer un processus et enfin de libérer les ressources.

- `squeue` : liste les jobs par file d'attente
- `sinfo` : donne l'utilisation des files d'attente
- `idle` : noeud libre
- `salloc -n 2 -p cpu` : allocation de 2 processeurs sur la file `cpu`
- `srun hostname` : retourne des informations sur les ressources (`jobid, ...`)
- `srun ./job.e` : lance le processus `job.e` sur les ressources réservées
- `exit` : libère les ressources

Utiliser les modules

L'utilisation des modules permet de compléter très simplement un certain nombre de variables d'environnement telles que **\$PATH, \$LD_LIBRARY_PATH**

Pour chaque application / bibliothèque disponible sur le cluster, vous trouverez donc un module à charger pour pouvoir l'utiliser. Les modules ont pour nom : <nom_application>/<version>

Les modules sont gérés de façon hiérarchique :

- les modules qui ne dépendent de rien
- les modules qui dépendent d'un compilateur
- les modules qui dépendent d'un couple (compilateur – MPI)

Ainsi

- si aucun module n'est chargé, la commande `module avail` donne la liste des modules qui ne dépendent de rien.
- Si vous chargez un compilateur, la commande `module avail` donne la liste des modules qui dépendent de ce compilateur.
- Si vous chargez un compilateur et un module MPI, la commande `module avail` donne la liste des modules qui dépendent de ce couple (compilateur – MPI)
- Lors du chargement d'un module, si la version n'est pas spécifiée alors c'est la version par défaut qui sera utilisée. Voici les * commandes de base de la commande module :

TABLEAU 1 – Commandes de base

module spider	liste tous les modules disponibles indépendamment des dépendances
module spider r/4.0	liste tous les modules nécessaires au fonctionnement du module r/4.0
module avail	liste tous les modules disponibles en fonction des modules déjà chargés
module list	liste tous les modules chargés par l'utilisateur
module load/unload openmpi/4.1	<p>charge/décharge le module openmpi/4.1</p> <ul style="list-style-type: none"> — Les commandes mpicc/mpif90 ou mpirun sont dans le PATH par défaut — Les libs adéquates sont chargés lors des opérations de compilation/exécution
module whatis < nom module >	description synthétique du module
module help	aide sur la syntaxe
module purge	décharge tous les modules
module load gcc/10.2 openmpi/4.1	pour un démarrage rapide vous pouvez charger les modules gcc/10.2 et openmpi/4.1

8.1 Utiliser R

R est installé avec le compilateur INTEL. Pour charger R vous devez utiliser la commande : `module load intel/2021.1 r/4.0`

Pour pouvoir installer vos paquets R localement sur votre home :

1. Créer un dossier sur votre compte : `mkdir ~/MyLibsR`
2. Créer le fichier `~/.Rprofile` contenant la ligne suivante : `.libPaths(« ~/MyLibsR »)`
3. Une fois R lancé, utiliser la commande : `install.packages(<pkg>,dependencies = TRUE,clean=TRUE, repos = <adresse du site>)`

Par exemple `install.packages("accuracy",dependencies = TRUE,clean=TRUE,repos = "http://mirror.ibcp.fr/pub/CRAN/)` L'installation se fera dans le dossier `~/MyLibsR`

8.2 Utiliser Python

Python est installé avec la distribution anaconda. Pour charger python vous devez utiliser la commande `module load anaconda/2020.11`

8.2.1 Anaconda

Vous pouvez utiliser une version installée en standard sur le cluster ou installer votre propre version sur votre compte.

Installer votre version d'anaconda

Vous pouvez télécharger [ici](#) l'installateur d'Anaconda et suivre la [procédure d'installation](#). La suite des explications de ce document est donnée en partant du principe que vous utilisez la distribution disponible sous forme de module. A vous d'adapter les explications si vous utilisez votre version d'anaconda.

Utiliser la version d'anaconda installée en standard sur le cluster

Une version a été installée pour vous. Pour l'utiliser : `module load anaconda/2020.11`

Environnements préexistants

Des environnements virtuels conda ont été créés. Pour avoir la liste des environnements virtuels existants : `conda env list`

Vous trouverez notamment 2 environnements virtuels disponibles pour les nœuds `gpu-intel` et `gpu-amd` pour les applications tensorflow (`tensorflow-gpu-env`) et py-torch (`pytorch-env`)

Note : ces environnements seront peut-être incomplets pour vos usages (packages supplémentaires nécessaires). Dans ce cas il vous faudra créer vos environnements virtuels sur votre compte

Configurer votre espace de travail pour faire des environnements dans votre compte

Pour pouvoir installer vos packages python localement sur votre compte :

- Créer un dossier sur votre compte : `mkdir -p ~/conda_env/.conda`
- Faire un lien symbolique vers `.conda` : `ln -s ~/conda_env/.conda ~/.conda`

Avertissement : Il peut être utile de supprimer le `.conda` existant ou d'en copier le contenu dans `~/conda_env/.conda`

- Créer une variable d'environnement : `export CONDA_ENVS_PATH=$HOME/conda_env/` (Vous pouvez ajouter cette variable dans votre `~/.bashrc` pour ne pas avoir à la saisir à chaque fois)
- Charger le module : `module load anaconda`
- Créer votre environnement virtuel en spécifiant le chemin : `conda create --prefix=${CONDA_ENVS_PATH}/test-3.6 python=3.6`
- Pour charger cet environnement : `source activate ${CONDA_ENVS_PATH}/test-3.6`
- Pour quitter cet environnement : `source deactivate`

8.2.2 Notebook

Préparer vos environnements virtuels afin qu'ils soient utilisables dans les notebooks

Si vous voulez utiliser vos environnements virtuels dans un notebook vous devez :

- Charger votre environnement virtuel (nommé `nom_env`) : `source activate ${CONDA_ENVS_PATH}/nom_env`
- Installer ipykernel dans cet environnement virtuel : `conda install ipykernel`
- Lier votre environnement virtuel à jupyter : `python -m ipykernel install --user --name=nom_env`
(L'environnement virtuel doit être actif)

Lancer un serveur jupyter sur des ressources dédiées

- Créer un script de soumission SLURM (`my_notebook.job`) pour votre jupyter notebook en décrivant les ressources dont vous avez besoin

```
#!/bin/bash
## nombre de noeuds
#SBATCH --nodes 1
##nombre de coeurs
#SBATCH --cpus-per-task 2
#durée de votre notebook
#SBATCH --time 02:00:00
#nomde votre notebook
#SBATCH --job-name my_jupyter_notebook
# on charge le module anaconda
module load anaconda/2020.11
#on lance le serveur jupyter (sans le navigateur) sur le port 8080
#on pouvait utiliser un autre port, de préférence supérieur à 8080
jupyter notebook --no-browser --port=8080 --ip=$(hostname -s)
```

- Soumettre votre job `sbatch my_notebook.job` : cette commande vous attribue un jobid et génère un fichier `slurm-jobid.out` où sont redirigées les sorties de votre job.
- Deux informations sont à récupérer dans le fichier : le nœud sur lequel s'exécute votre job (`nodexx` ou `gpu0x`) et le token d'authentification au notebook. Vous pouvez éditer le fichier `slurm-jobid.out` ou utiliser la commande `grep token slurm-jobid.out` (remplacer `jobid` par la valeur rendue par la commande `sbatch` précédente) La ligne qui nous intéresse ressemble à

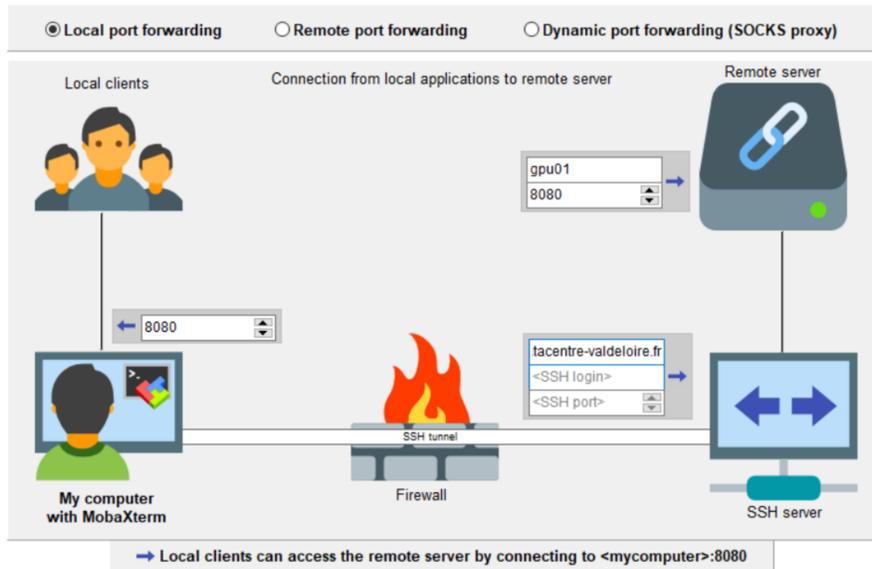
<http://gpu01:8080/?token=817f5205e74d91f0f670d567945e45c4514a6ef1aed49e93>

Dans cet exemple le nœud sur lequel s'exécute jupyter est `gpu01`, le token est `817f5205e74d91f0f670d567945e45c4514a6ef1aed49e93`

Créer un tunnel ssh entre votre poste et le nœud où s'exécute votre notebook

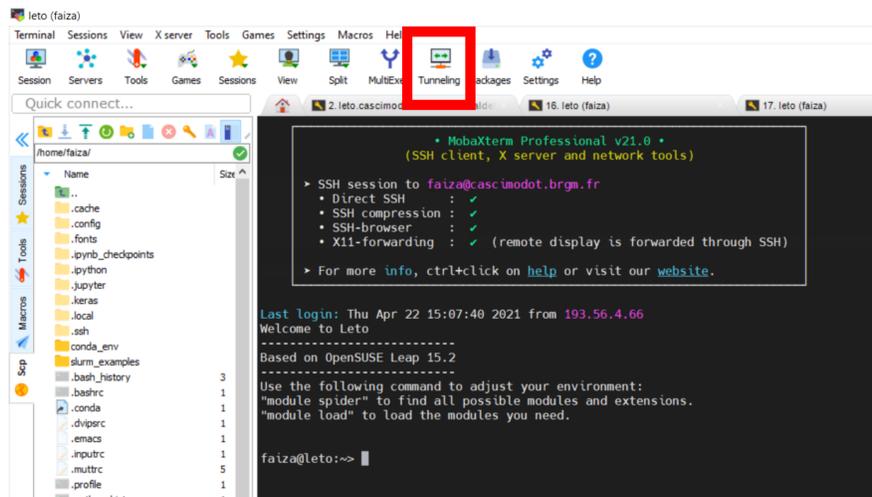
Le serveur jupyter est un serveur web, en écoute sur un port sur un nœud du cluster. Pour y accéder depuis votre poste vous devez établir un tunnel construit avec le protocole SSH entre un client (votre poste) et un serveur (Leto) en faisant correspondre un port de votre poste au port distant utilisé par le notebook jupyter, et ce à travers une connexion SSH.

Pour mieux comprendre ce qu'est un tunnel ssh vous pouvez utiliser cette représentation proposée par `mobaxterm` :

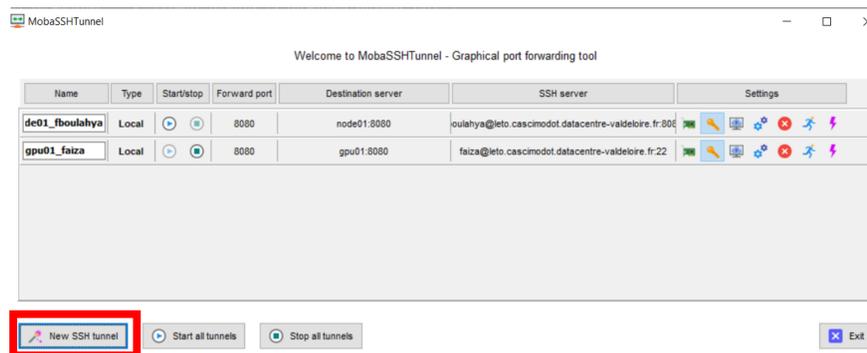


Avec MobaXterm

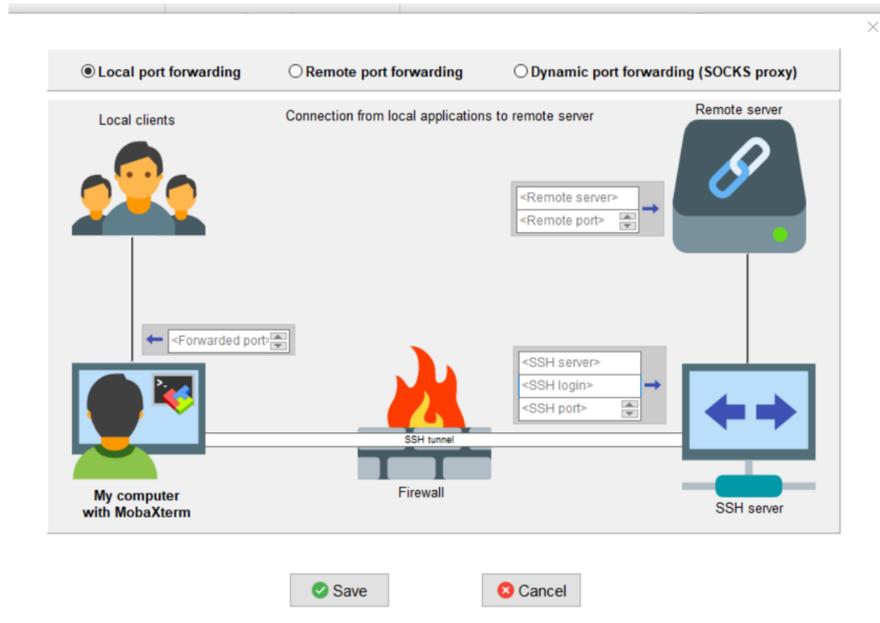
Dans MobaXterm démarrer MobaSSHTunnel en cliquant sur l'icône Tunneling



La fenêtre MobaSSHTunnel s'ouvre. Cliquer sur New SSH Tunnel



Une fenêtre pour configurer votre tunnel s'ouvre.



Laisser Local port forwarding coché et renseigner les champs de la manière suivante :

- Forwarded port : 8080 (il s'agit du port local de votre machine)
- SSH server : leto.cascimodot.datacentre-valdeleire.fr
- SSH login : votre login sur Leto
- Remote server : le nœud où s'exécute votre notebook
- Remote port : 8080

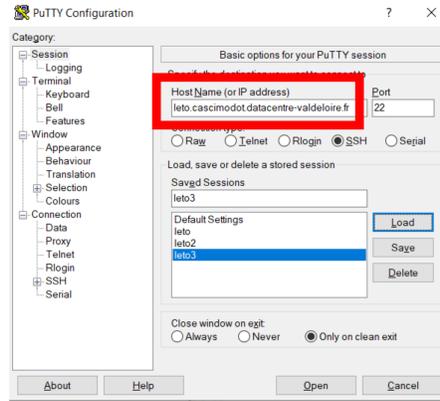
Cliquer sur Save. Dans la fenêtre MobaSSHTunnel une ligne a été ajoutée. Vous pouvez donner un nom à ce tunnel. Vous devez ensuite ajouter votre clé privée ssh nécessaire à la connexion sur Leto. Pour cela, cliquer sur l'icône clé au bout de la ligne



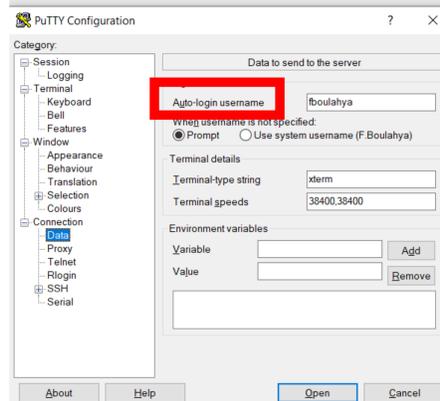
Vous n'avez plus qu'à cliquer sur la petite icône start pour que le tunnel démarre. Nom Start Clé

Avec putty

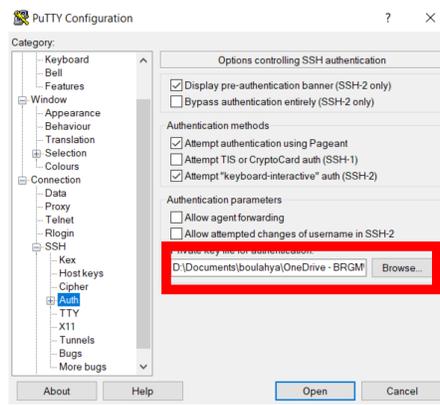
Vous devez renseigner un profil de connexion putty de la manière suivante : Dans session / Host Name : leto.cascimodot.datacentre-valdeleire.fr



Dans Connection / Data / auto-login username : votre login de connexion à Leto



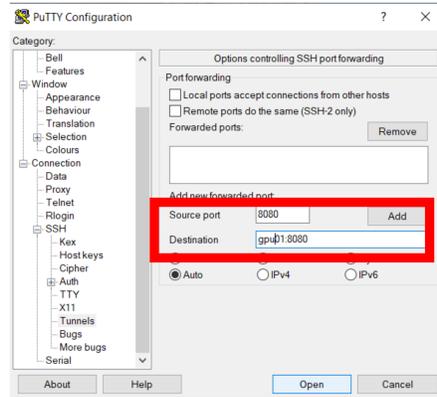
Dans Connection / SSH / Auth renseigner votre clé privée



Dans Connection / SSH / Tunnels / Source port : 8080 (port local de votre machine)

Dans Connection / SSH / Tunnels / Destination : node :8080 (où node est à remplacer par le nœud où s'exécute votre notebook)

Cliquer sur add



Enfin, cliquer sur open. Un terminal s'ouvre sur Leto

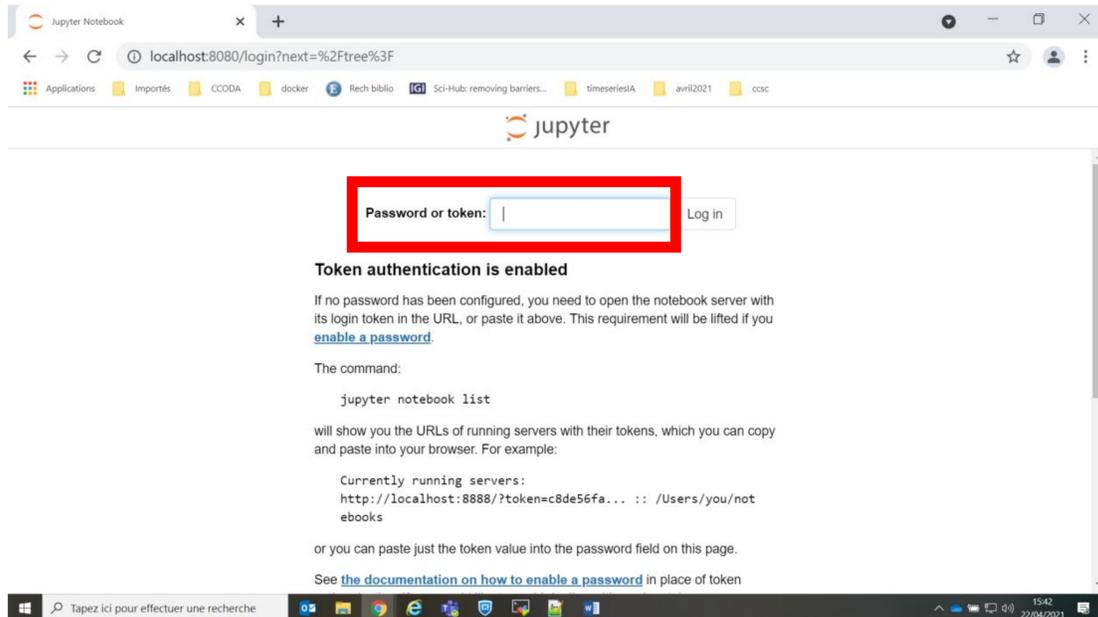
Depuis Linux

La commande pour créer un tunnel ssh vers le nœud nodexx pour l'utilisateur toto est :

```
ssh -N -L 8080:nodexx:8080 toto@leto.cascimodot.datacentre-valdeloire.fr
```

Le notebook

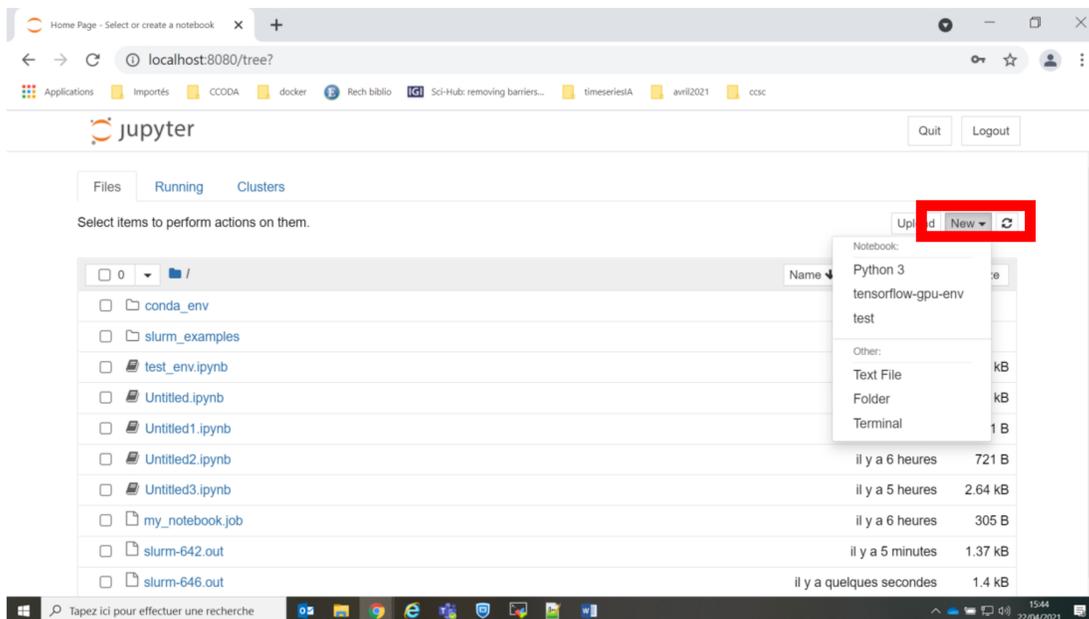
Dans votre navigateur internet taper l'adresse <http://localhost:8080>



Dans le champ token saisir le token que vous avez récupéré dans votre fichier slurm-jobid.out Vous obtenez votre page jupyter listant le contenu de votre compte (de votre home directory)

Sélectionner un environnement dans le notebook

Cliquer sur le bouton new et sélectionner l'environnement virtuel qui vous convient. Si vous n'avez pas lié d'environnements virtuels à jupyter vous ne verrez que Python 3



8.2.3 Carte GPU

Vérification avant de pouvoir utiliser les gpu

Pour pouvoir utiliser les gpu vous devez être dans le groupe linux video.

Pour s'assurer que vous êtes dans ce groupe taper la commande `groups`. Si video n'apparait pas, vous pouvez nous contacter.

Ressources GPU

Dans votre script de soumission SLURM ajouter une des 2 partitions gpu avec une de ces 2 lignes

```
#SBATCH -p gpu-intel
#SBATCH -p gpu-amd
```

Ajouter également le nombre de gpus (x) que vous voulez utiliser

```
#SBATCH --gres=gpu:x
```

Note : Rappel pour chaque nœud :

- partition **gpu-intel** : 4 x Nvidia Tesla V100 (Nvlink)
- partition **gpu-amd** : 3 x Nvidia Tesla V100 (PCIe)