SSH

Nikita Neveditsin, SMU, 2019

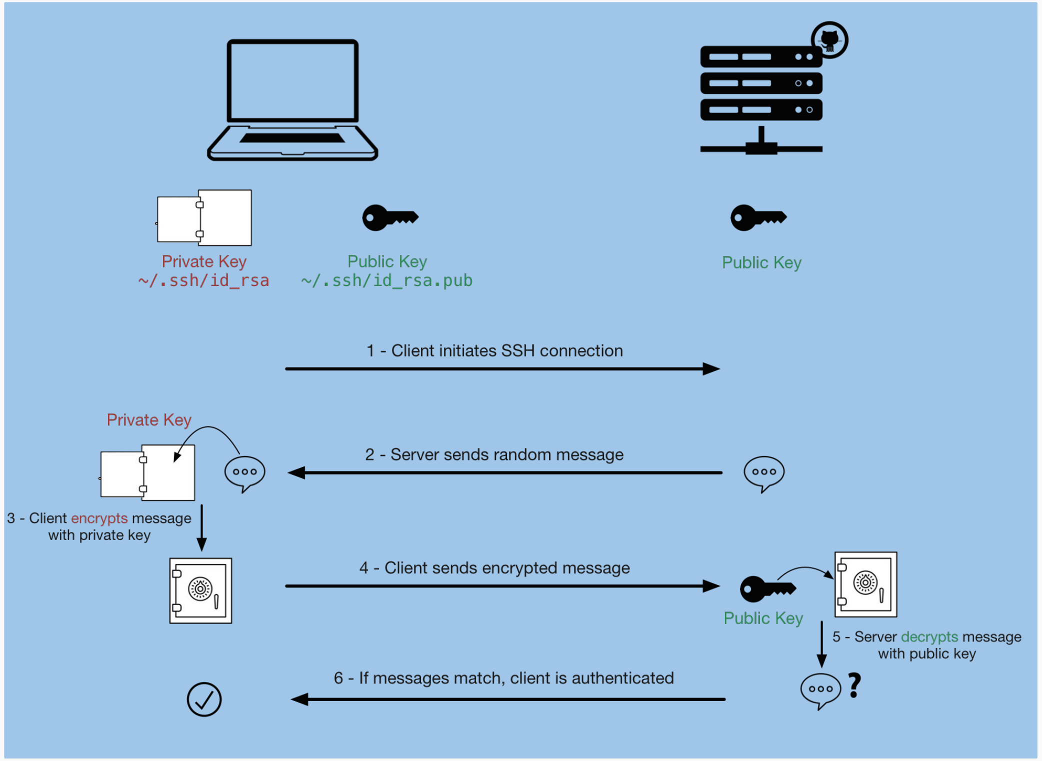
nikita.neveditsin@smu.ca

What is SSH?

SSH protocol (Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an *unsecured* network. SSH provides a secure channel over an unsecured network in a client-server architecture, connecting an SSH client application with an SSH server. [1]). Version 2 of SSH is now used. Default port: 22

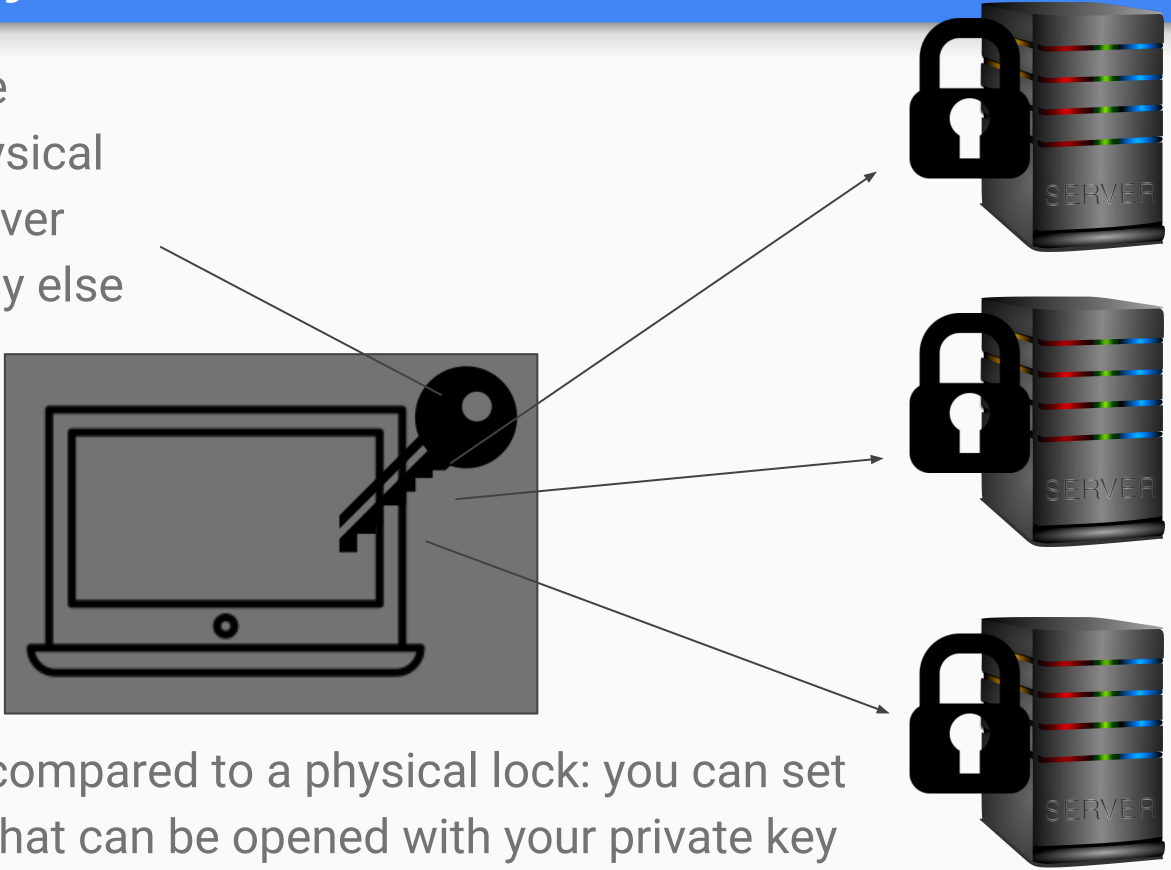
Public key authentication

ssh client can be authenticated either using a password or a public key


Private Key
~/.ssh/id_rsa
Public Key
~/.ssh/id_rsa.pub
Public Key
1 - Client initiates SSH connection
Private Key
2 - Server sends random message
3 - Client encrypts message with private key
4 - Client sends encrypted message
Public Key
5 - Server decrypts message with public key
6 - If messages match, client is authenticated
?

Public key authentication

Private Key can be compared to a physical key: you should never give it to somebody else



Public Key can be compared to a physical lock: you can set "locks" on servers that can be opened with your private key

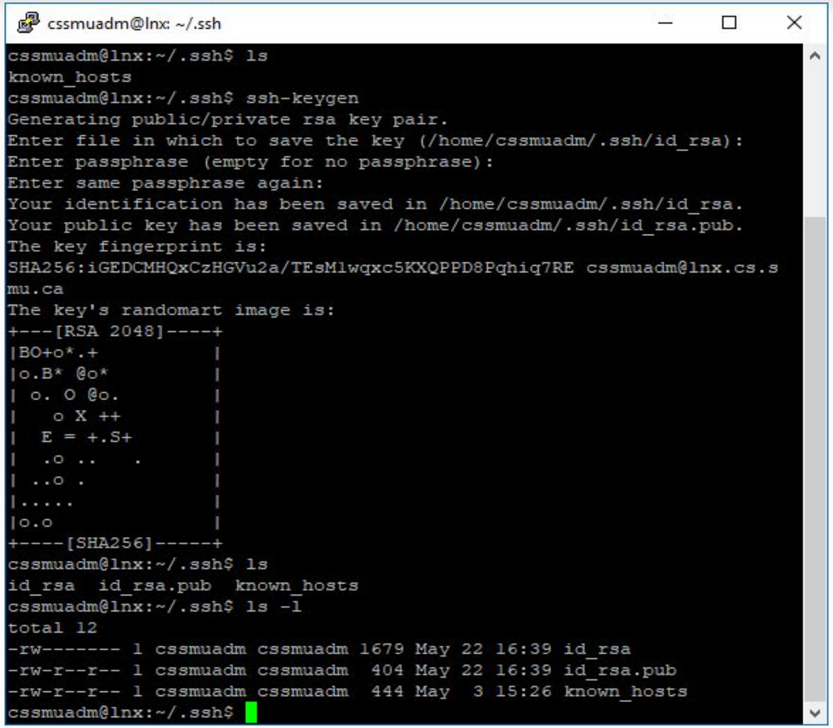
Connect from lnx to dev: step 1

Step 1:

ssh client should generate a key pair using ssh-keygen command (usually a passphrase is not used). Public (id_rsa.pub) and private (id_rsa) keys will be generated and stored in ~/.ssh directory. If a key pair is generated, skip this step.

Note: public key can easily be retrieved from private key using the following command:

ssh-keygen -y -f ~/.ssh/id_rsa > id_rsa.pub



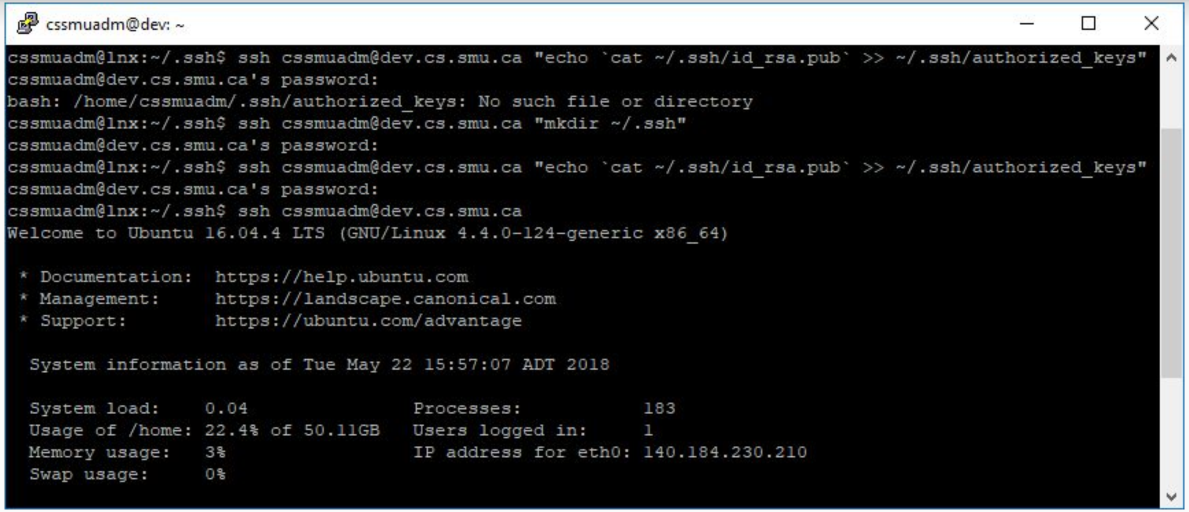
```
cssmuadm@lnx:~/.ssh$ ls
known_hosts
cssmuadm@lnx:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cssmuadm/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cssmuadm/.ssh/id_rsa.
Your public key has been saved in /home/cssmuadm/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:iGEDCMHQxCzHGVu2a/TEsM1wqxc5KXQPPD8Pqhiq7RE cssmuadm@lnx.cs.s
mu.ca
The key's randomart image is:
+---[RSA 2048]----+
|BO+o*.+          |
|o.B* @o*         |
| o. O @o.        |
|   o X ++        |
|  E = +.S+       |
|  .o ..    .     |
| ..o .           |
|.....            |
|o.o              |
+----[SHA256]-----+
cssmuadm@lnx:~/.ssh$ ls
id_rsa  id_rsa.pub  known_hosts
cssmuadm@lnx:~/.ssh$ ls -l
total 12
-rw------- 1 cssmuadm cssmuadm 1679 May 22 16:39 id_rsa
-rw-r--r-- 1 cssmuadm cssmuadm  404 May 22 16:39 id_rsa.pub
-rw-r--r-- 1 cssmuadm cssmuadm  444 May  3 15:26 known_hosts
cssmuadm@lnx:~/.ssh$
```

Connect from lnx to dev: step 2

Step 2:

You should add your public key to the remote server's ~/.ssh/authorized_keys file. **If .ssh directory does not exist, create it:**

- Use the one-line command as in the example
- Or copy the public key first to the remote server with scp then append ~/.ssh/authorized_keys with it
- Or edit ~/.ssh/authorized_keys manually



```
cssmuadm@lnx:~/.ssh$ ssh cssmuadm@dev.cs.smu.ca "echo `cat ~/.ssh/id_rsa.pub` >> ~/.ssh/authorized_keys"
cssmuadm@dev.cs.smu.ca's password:
bash: /home/cssmuadm/.ssh/authorized_keys: No such file or directory
cssmuadm@lnx:~/.ssh$ ssh cssmuadm@dev.cs.smu.ca "mkdir ~/.ssh"
cssmuadm@dev.cs.smu.ca's password:
cssmuadm@lnx:~/.ssh$ ssh cssmuadm@dev.cs.smu.ca "echo `cat ~/.ssh/id_rsa.pub` >> ~/.ssh/authorized_keys"
cssmuadm@dev.cs.smu.ca's password:
cssmuadm@lnx:~/.ssh$ ssh cssmuadm@dev.cs.smu.ca
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-124-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue May 22 15:57:07 ADT 2018

  System load:    0.04               Processes:           183
  Usage of /home: 22.4% of 50.11GB   Users logged in:     1
  Memory usage:   3%                 IP address for eth0: 140.184.230.210
  Swap usage:     0%
```

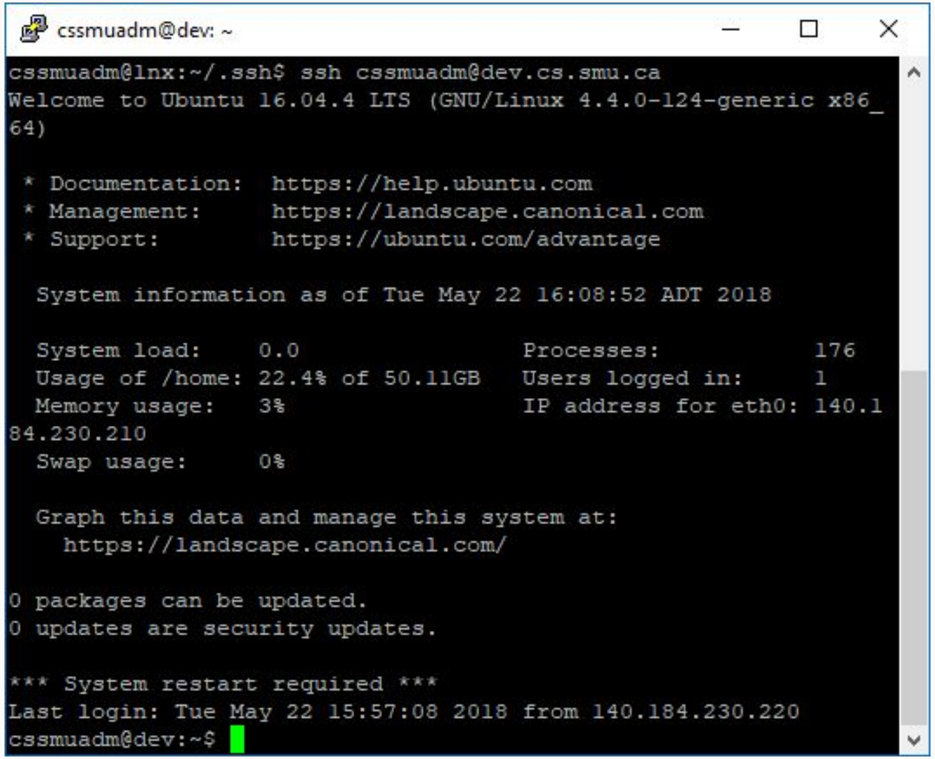
Note: the command between backticks (`cat ~/.ssh/id_rsa.pub`) is evaluated before the main command (echo)

Connect from lnx to dev: step 3

Step 3

connect without any passwords:

ssh your_username@dev.cs.smu.ca

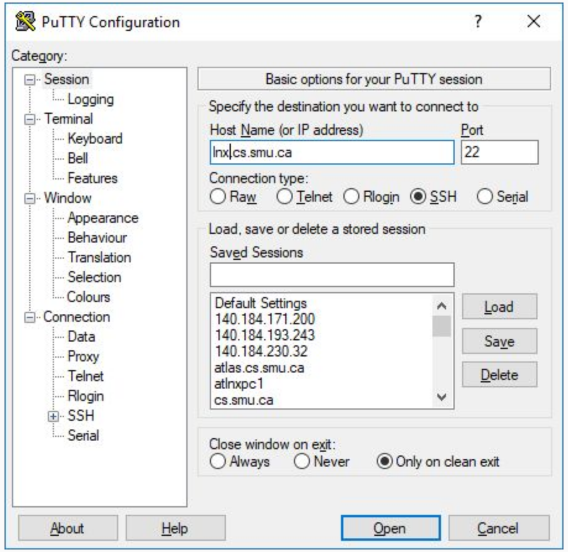


```
cssmuadm@lnx:~/.ssh$ ssh cssmuadm@dev.cs.smu.ca
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-124-generic x86_
64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue May 22 16:08:52 ADT 2018

  System load:    0.0                Processes:           176
  Usage of /home: 22.4% of 50.11GB   Users logged in:     1
  Memory usage:   3%                 IP address for eth0: 140.1
84.230.210
  Swap usage:     0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Tue May 22 15:57:08 2018 from 140.184.230.220
cssmuadm@dev:~$
```

Exercise

*Set up a public key authentication from **lnx.cs.smu.ca** to **dev.cs.smu.ca***

Connect from Windows to remote Linux



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): lnx.cs.smu.ca

Port: 22

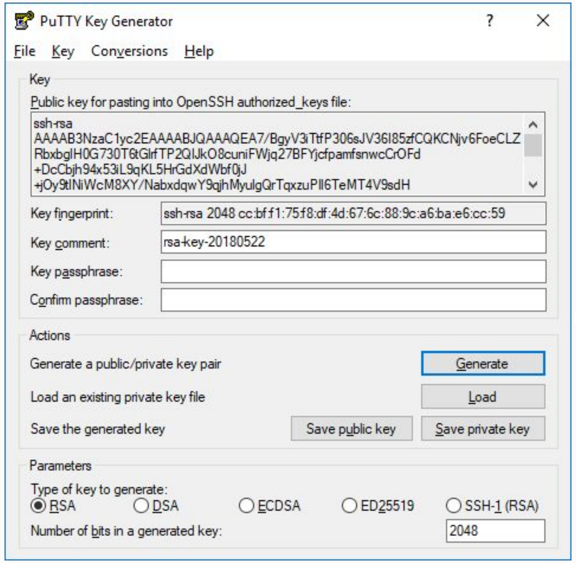
Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
140.184.171.200
140.184.193.243
140.184.230.32
atlas.cs.smu.ca
atlnxpc1
cs.smu.ca

Close window on exit: Always Never [✓ Only on clean exit]



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEA7/BgyV3iTtfP306sJV36I85zfCQKCNjv6FoeCLZ
RbxbglH0G730T6tGlrfTP2QIJkO8cuniFWjq27BFYjcfpamfsnwcCrOFd
+DcCbjh94x53iL9qKL5HrGdXdWbf0jJ
+jOy9tINiWcM8XY/NabxdqwY9qjhMyulgQrTqxzuPlI6TeMT4V9sdH
```

Key fingerprint: ssh-rsa 2048 cc:bf:f1:75:f8:df:4d:67:6c:88:9c:a6:ba:e6:cc:59

Key comment: rsa-key-20180522

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

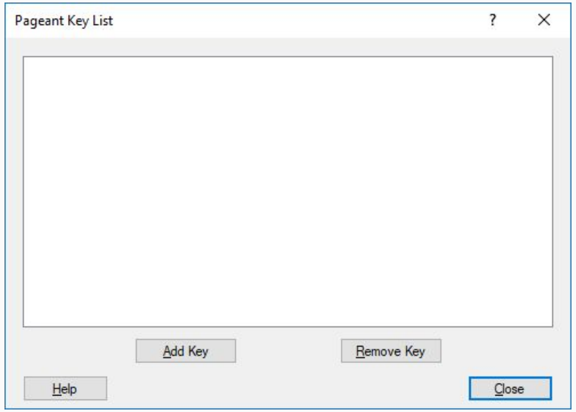
Load an existing private key file

Save the generated key

Parameters

Type of key to generate: [✓ RSA] DSA ECDSA ED25519 SSH-1 (RSA)

Number of bits in a generated key: 2048

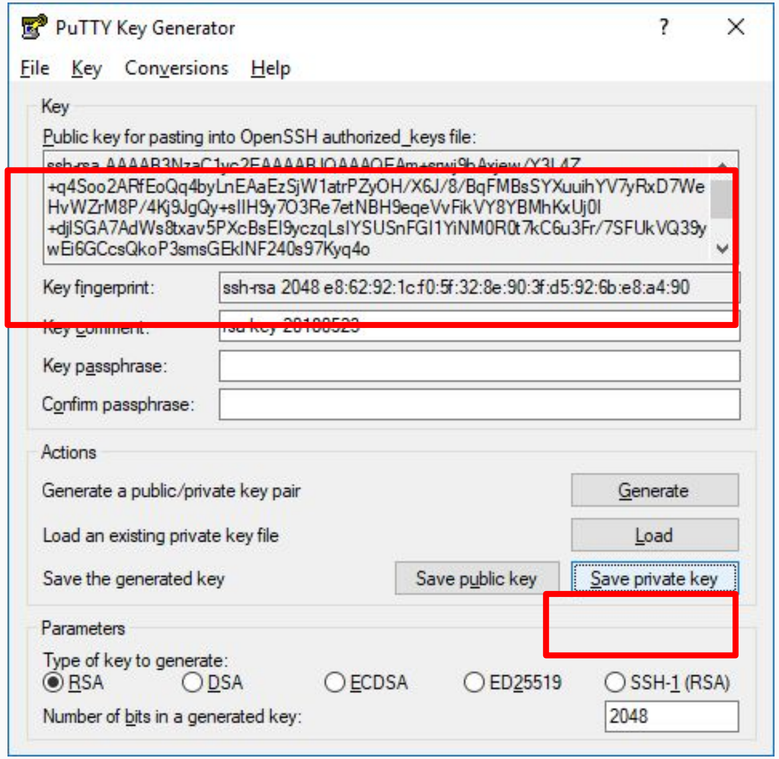


Connect from Windows to remote Linux: step 1

Step 1:

Open **PuTTYGen**

- Generate RSA key
- Save a private key in some directory that nobody else have access to
- Copy a “**public key for pasting into OpenSSH authorized_keys file**”


PuTTY Key Generator

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAm+snwj9hAxiew/Y3L4Z
+q4Soo2ARfEoQq4byLnEAaEzSjW1atrPZyOH/X6J/8/BqFMBsSYXuuihYV7yRxD7We
HvWZrM8P/4Kj9JgQy+sIIH9y7O3Re7etNBH9eqeVvFikVY8YBMhKxUj0I
+djISGA7AdWs8txav5PXcBsEI9yczqLsIYSUSnFGI1YiNM0R0t7kC6u3Fr/7SFUkVQ39y
wEi6GCcsQkoP3smsGEkINF240s97Kyq4o
```

Key fingerprint: ssh-rsa 2048 e8:62:92:1c:f0:5f:32:8e:90:3f:d5:92:6b:e8:a4:90

Key comment: rsa-key-20180523

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

Parameters

Type of key to generate: [✓ RSA] DSA ECDSA ED25519 SSH-1 (RSA)

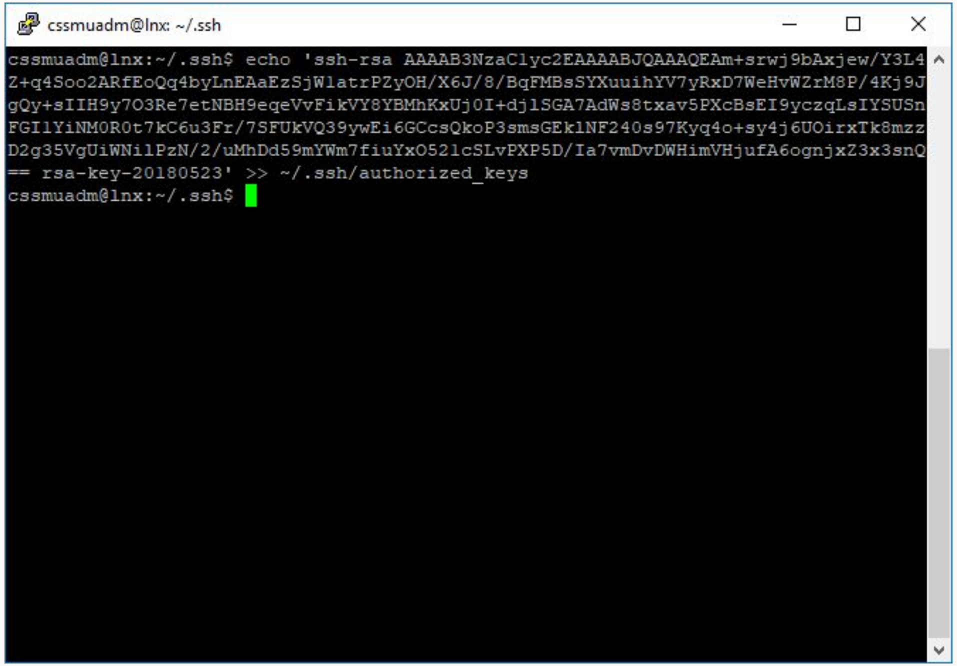
Number of bits in a generated key: 2048

Connect from Windows to remote Linux: step 2

Step 2:

Open **PuTTY**

- Connect to the remote server
- Paste your public key from Step 1 to **authorized_keys** file



cssmuadm@lnx: ~/.ssh

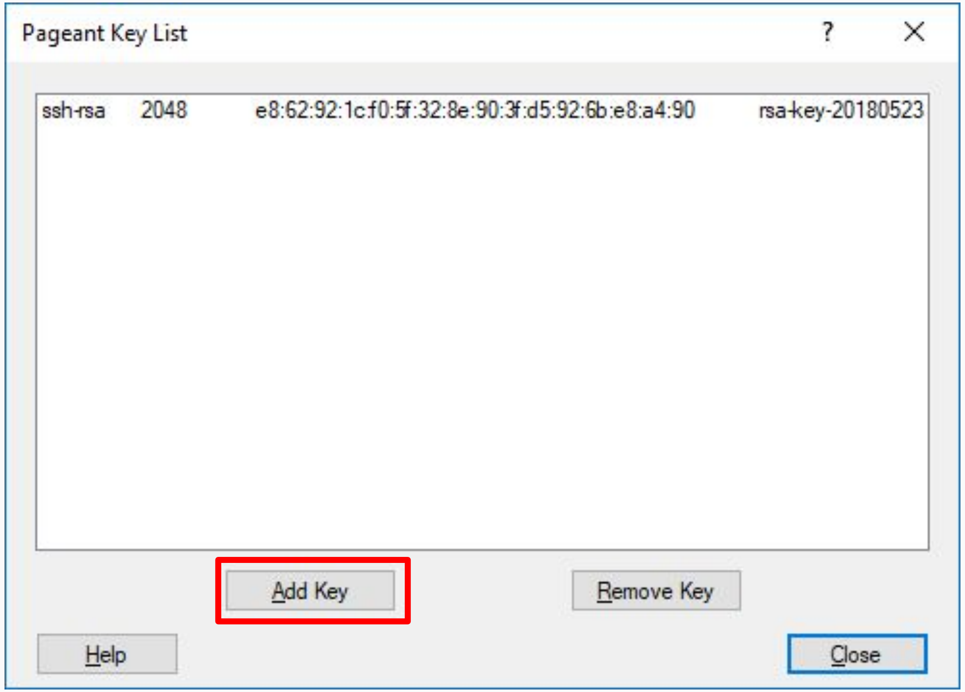
```
cssmuadm@lnx:~/.ssh$ echo 'ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAm+srwj9bAxjew/Y3L4
Z+q4Soo2ARfEoQq4byLnEAaEzSjWlatrPZyOH/X6J/8/BqFMBsSYXuuihYV7yRxD7WeHvWZrM8P/4Kj9J
gQy+sIIH9y7O3Re7etNBH9eqeVvFikVY8YBMhKxUj0I+dj1SGA7AdWs8txav5PXcBsEI9yczqLsIYSUSn
FGI1YiNM0R0t7kC6u3Fr/7SFUkVQ39ywEi6GCcsQkoP3smsGEk1NF240s97Kyq4o+sy4j6UOirxTk8mzz
D2g35VgUiWNilPzN/2/uMhDd59mYWm7fiuYxO521cSLvPXP5D/Ia7vmDvDWHimVHjufA6ognjxZ3x3snQ
== rsa-key-20180523' >> ~/.ssh/authorized_keys
cssmuadm@lnx:~/.ssh$
```

Connect from Windows to remote Linux: step 3

Step 3:

Open **Pageant**

- Press “Add Key” and add the private key from Step 1
- Close the window. Pageant will be running in background



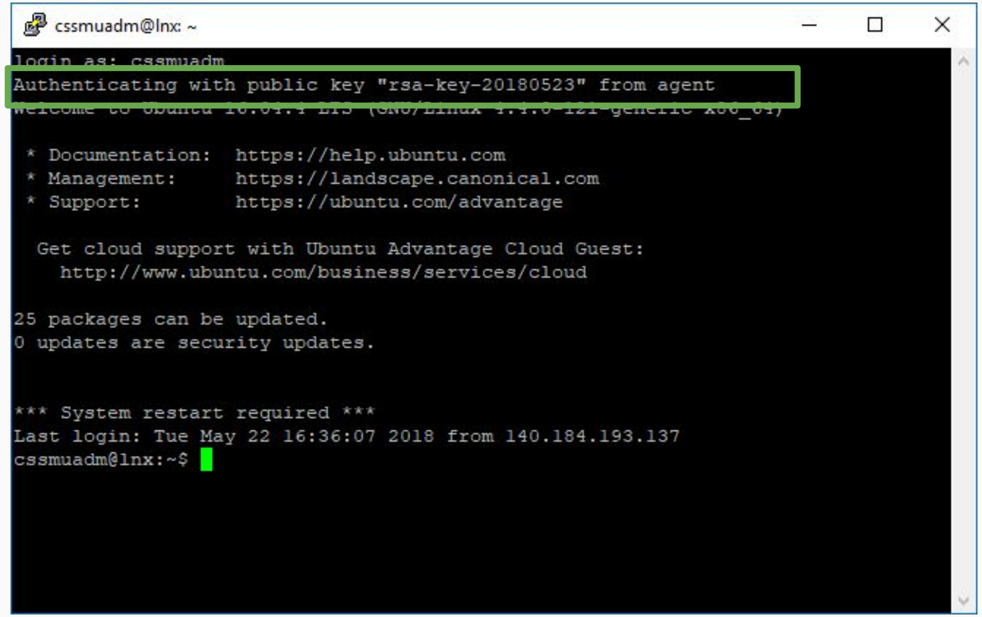
```
ssh-rsa    2048    e8:62:92:1c:f0:5f:32:8e:90:3f:d5:92:6b:e8:a4:90    rsa-key-20180523
```

Connect from Windows to remote Linux: step 4

Step 4:

Open **PuTTY** again

- Enter your username
- You are authenticated with public key from agent (Pageant) now

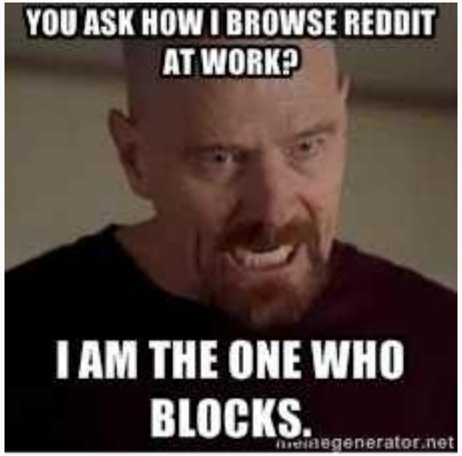


```
login as: cssmuadm
Authenticating with public key "rsa-key-20180523" from agent
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-121-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

25 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Tue May 22 16:36:07 2018 from 140.184.193.137
cssmuadm@lnx:~$
```

SSH Tunnels

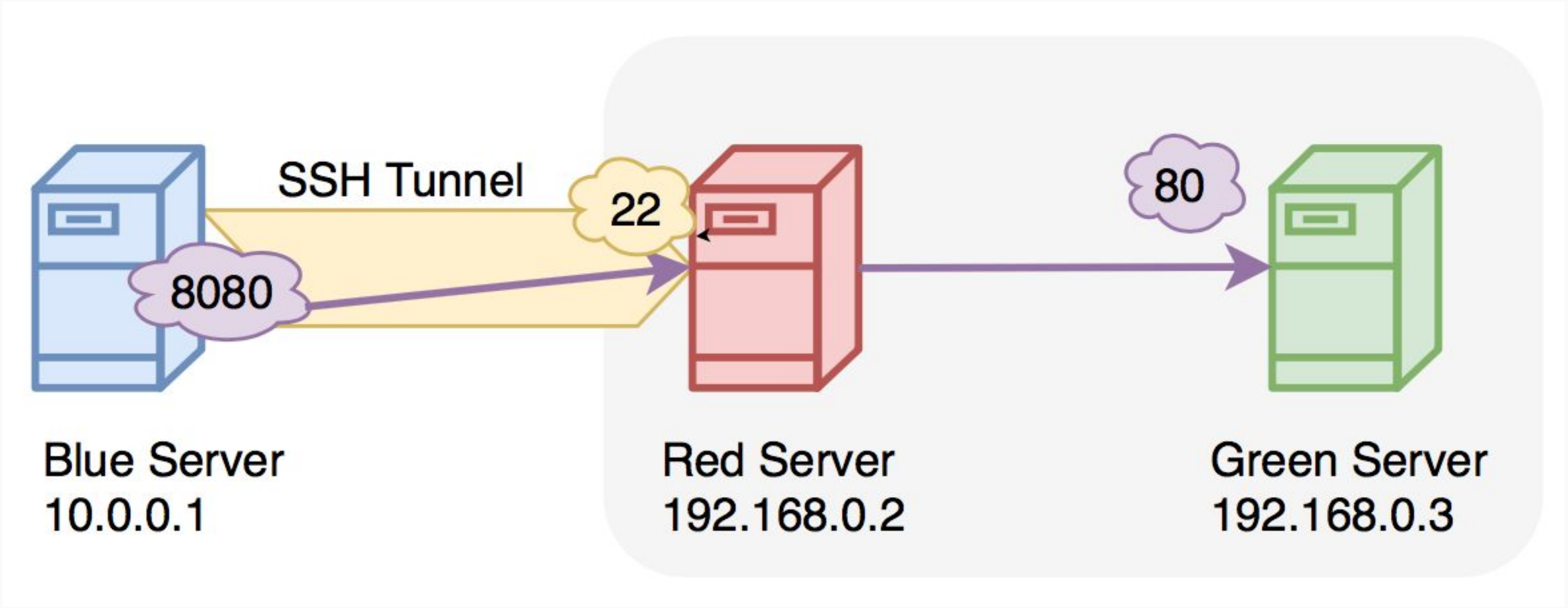
Why do we need them? Use cases:

- Access services from the remote server that listen on localhost (e.g., MySQL at **dev.cs.smu.ca**)
- Access blocked websites from your local machine if you have access to a server that can access that websites
- Secure traffic from your apps



[1]

SSH Tunnels (cont-d)



SSH Tunnels: scenario 1

Scenario:

- MySQL on dev.cs.smu.ca listens on **localhost** only (can be accessed only from the server itself)
- We need to connect to MySQL on dev.cs.smu.ca from
  - Either lnx.cs.smu.ca
  - Or Windows machine (using some MySQL client like HeidiSQL or JDBC)

SSH Tunnel between 2 *nix machines

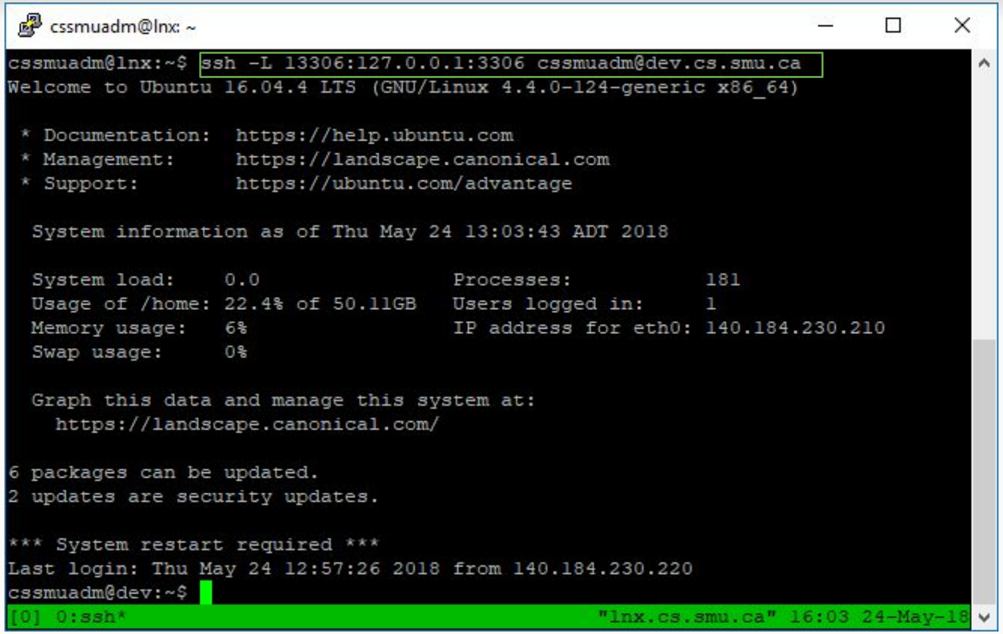
Step 1:

- Run tmux
- Type

ssh -L 13306:127.0.0.1:3306 user@dev.cs.smu.ca

- Detach from the session with **Ctrl+B d**
- Type mysql -uusername -p --port=13306 --host=127.0.0.1

NOTES: you can use any local available port instead of 13306. If somebody already uses port 13306 (or other), you cannot use the same port!



```
cssmuadm@lnx:~$ ssh -L 13306:127.0.0.1:3306 cssmuadm@dev.cs.smu.ca
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-124-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu May 24 13:03:43 ADT 2018

  System load:     0.0               Processes:           181
  Usage of /home: 22.4% of 50.11GB   Users logged in:     1
  Memory usage:    6%                IP address for eth0: 140.184.230.210
  Swap usage:      0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

6 packages can be updated.
2 updates are security updates.

*** System restart required ***
Last login: Thu May 24 12:57:26 2018 from 140.184.230.220
cssmuadm@dev:~$
[0] 0:ssh*                                   "lnx.cs.smu.ca" 16:03 24-May-18
```

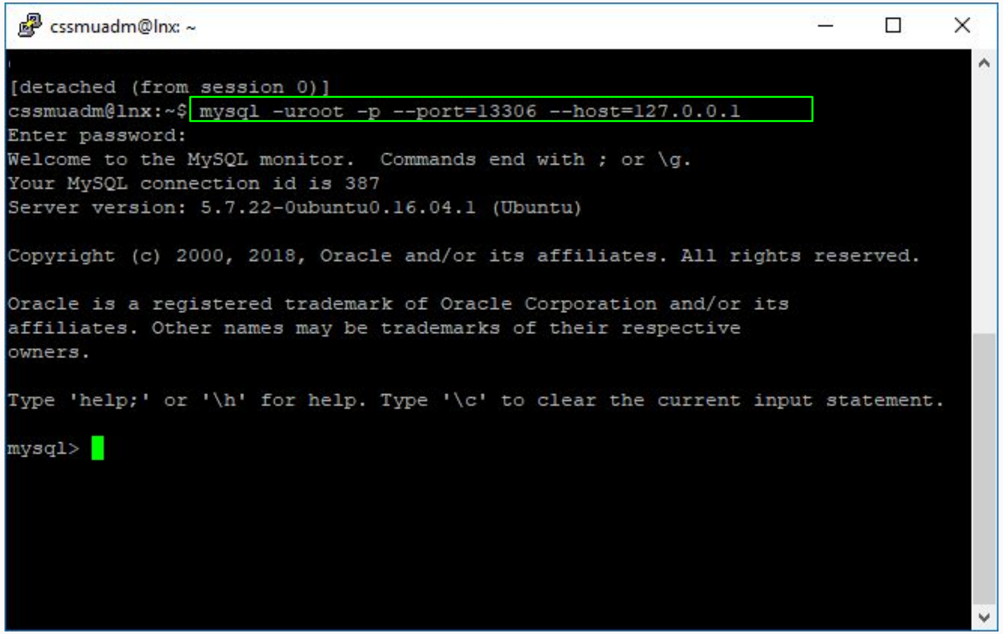
SSH Tunnel between 2 *nix machines (cont-d)

Explained:

ssh -L 13306:127.0.0.1:3306 user@dev.cs.smu.ca

- **-L** means LOCAL port 13306
- **127.0.0.1:3306** means REMOTE ip:port

Result: SSH opens LOCAL port 13306 and "maps" it to 127.0.0.1:3306 on dev.cs.smu.ca. So, when you connect to 127.0.0.1:13306 on the CLIENT machine (in our case lnx.cs.smu.ca) it connects to 127.0.0.1:3306 on dev.cs.smu.ca



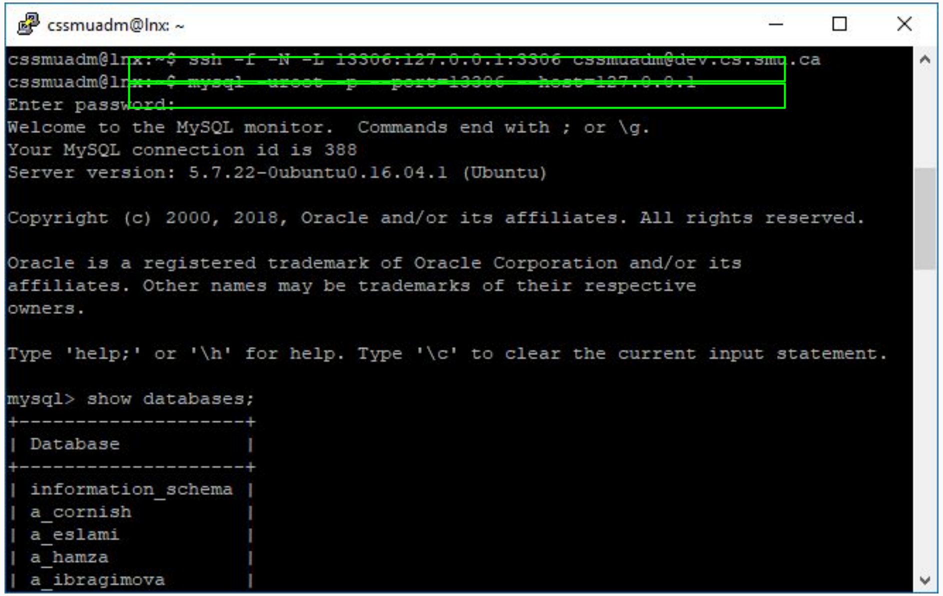
```
[detached (from session 0)]
cssmuadm@lnx:~$ mysql -uroot -p --port=13306 --host=127.0.0.1
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 387
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

SSH Tunnel between 2 *nix machines (cont-d)

Without tmux:

ssh -f -N -L 13306:127.0.0.1:3306 user@dev.cs.smu.ca

- -f Requests ssh to go to background just before command execution.
- -N Do not execute a remote command. This is useful for just forwarding ports.

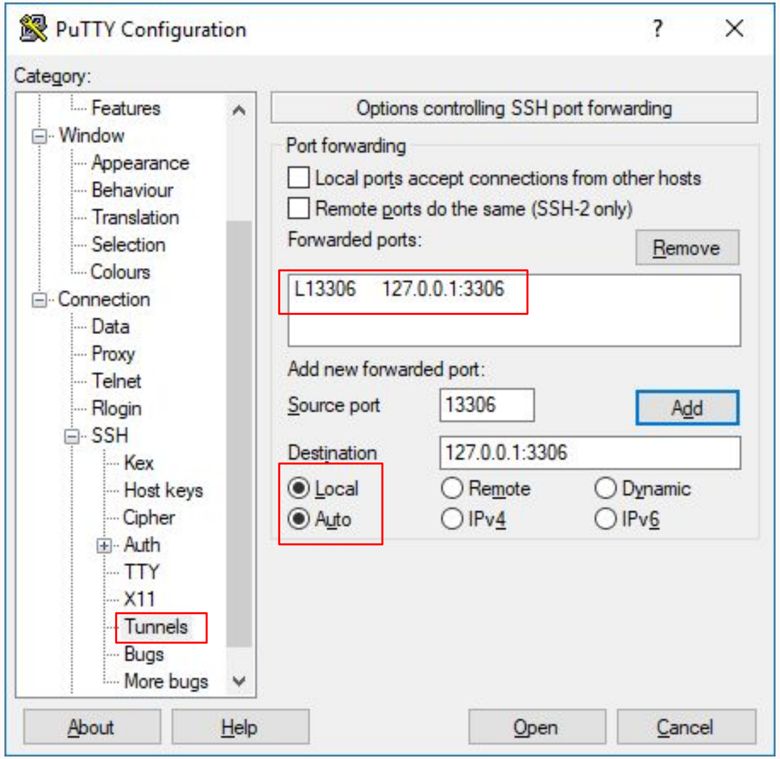


```
cssmuadm@lnx:~$ ssh -f -N -L 13306:127.0.0.1:3306 cssmuadm@dev.cs.smu.ca
cssmuadm@lnx:~$ mysql -uroot -p --port=13306 --host=127.0.0.1
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 388
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| a_cornish          |
| a_eslami           |
| a_hamza            |
| a_ibragimova       |
```

SSH Tunnel between Windows and Linux

Steps:

- Run PuTTY, enter the server's address (dev.cs.smu.ca). Go to the **Tunnels** tab. Add **L13306 127.0.0.1:3306** to Forwarded ports and Open session. Leave the terminal window opened.
- Run any MySQL client and connect to **127.0.0.1:13306**



Options controlling SSH port forwarding

Port forwarding

[ ] Local ports accept connections from other hosts

[ ] Remote ports do the same (SSH-2 only)

Forwarded ports:

```
L13306     127.0.0.1:3306
```

Add new forwarded port:

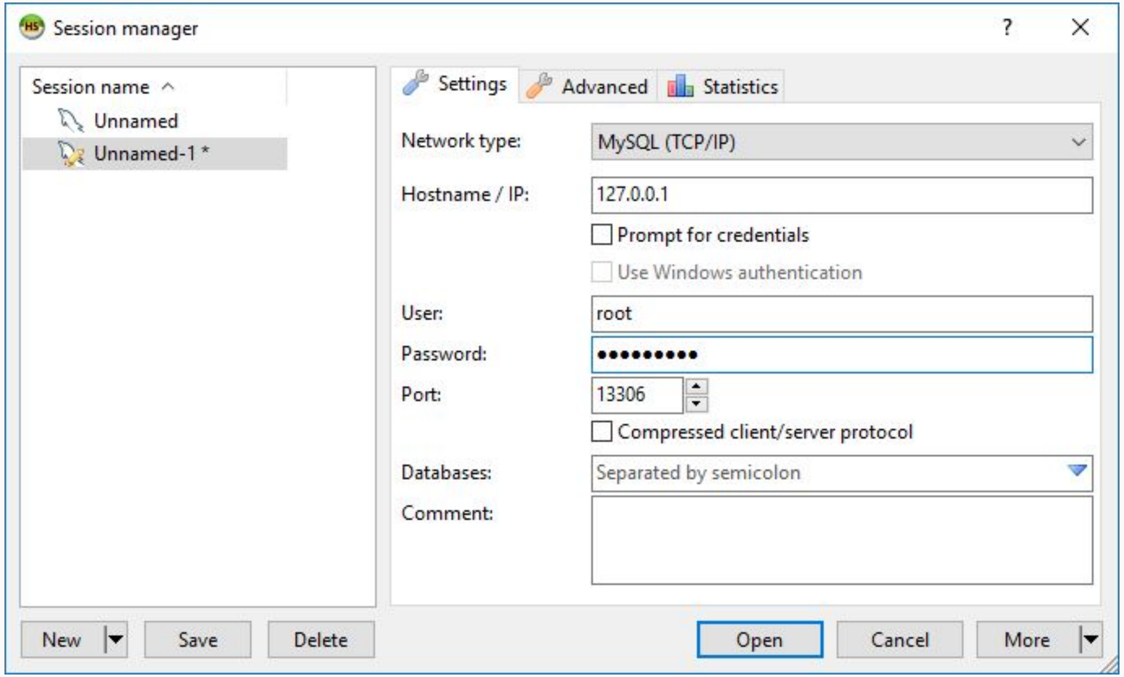
Source port: 13306

Destination: 127.0.0.1:3306

[✓ Local] Remote Dynamic

[✓ Auto] IPv4 IPv6

SSH Tunnel between Windows and Linux (cont-d)



Network type: MySQL (TCP/IP)

Hostname / IP: 127.0.0.1

[ ] Prompt for credentials

[ ] Use Windows authentication

User: root

Password: ●●●●●●●●●●

Port: 13306

[ ] Compressed client/server protocol

Databases: Separated by semicolon

Comment:

Exercise

Set up port forwarding from **lnx.cs.smu.ca** to MySQL server on **dev.cs.smu.ca** and connect to your MySQL database using any MySQL client

NOTE: use port = 1MMDD where MM-your month of birth, DD-day of birth

Exercise

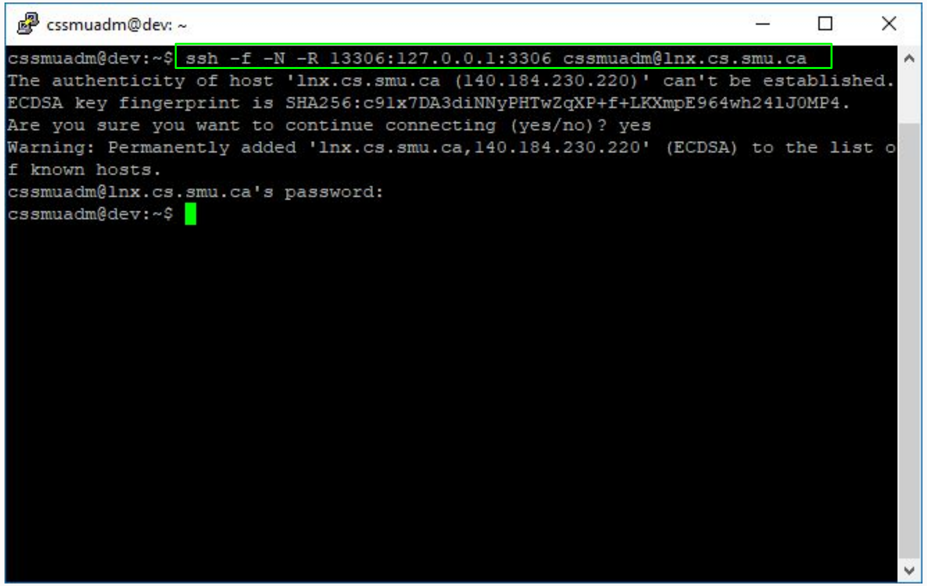
Using PuTTY on your Windows machine or command line on your Mac set up port forwarding to the **smu.ca** website (port 80) so that you can access that website from your browser by typing **http://127.0.0.1:1234**

SSH tunnels: remote port forwarding

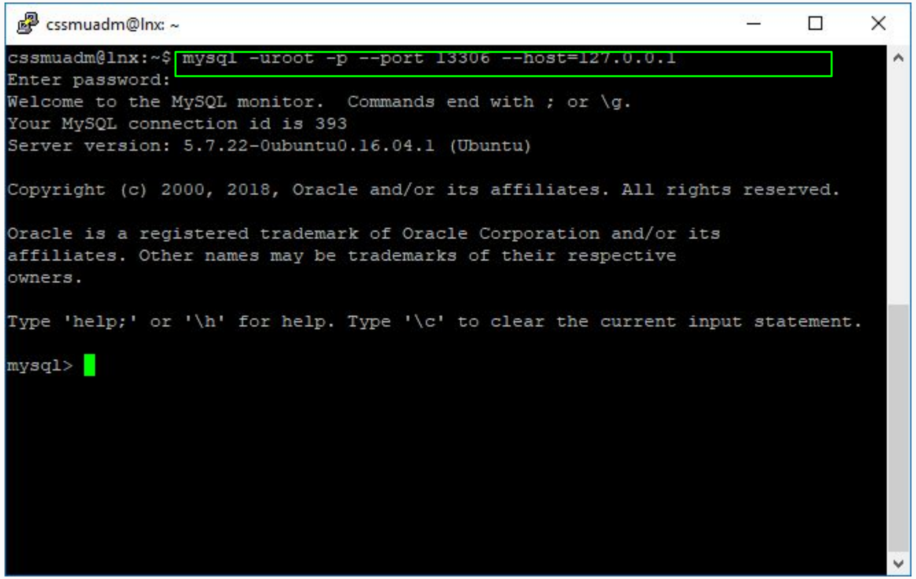
You can forward client’s port to the remote server. Just use **-R** instead of **-L**:

ssh -R 13306:127.0.0.1:3306 user@lnx.cs.smu.ca (the command is executed on **dev.cs.smu.ca**)

Will “map” 127.0.0.1:3306 (MySQL server port) from dev.cs.smu.ca to port 13306 at lnx.cs.smu.ca



```
cssmuadm@dev:~$ ssh -f -N -R 13306:127.0.0.1:3306 cssmuadm@lnx.cs.smu.ca
The authenticity of host 'lnx.cs.smu.ca (140.184.230.220)' can't be established.
ECDSA key fingerprint is SHA256:c91x7DA3diNNyPHTwZqXP+f+LKXmpE964wh241J0MP4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'lnx.cs.smu.ca,140.184.230.220' (ECDSA) to the list o
f known hosts.
cssmuadm@lnx.cs.smu.ca's password:
cssmuadm@dev:~$
```



```
cssmuadm@lnx:~$ mysql -uroot -p --port 13306 --host=127.0.0.1
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 393
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Reverse SSH

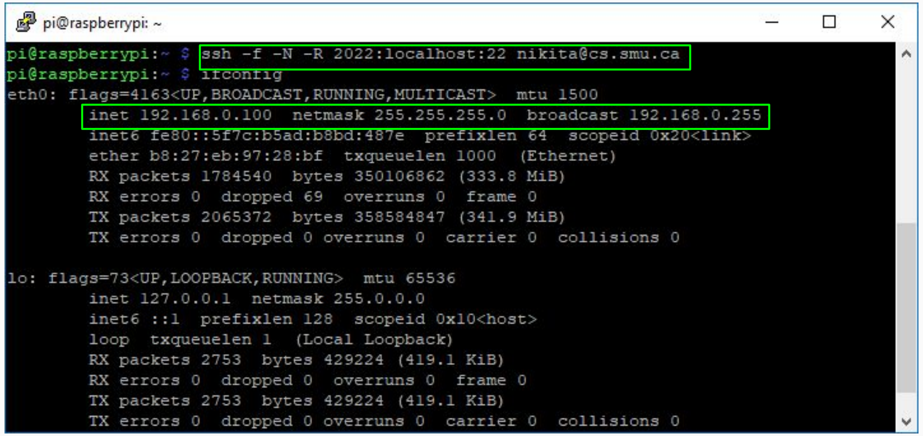
Why do we need Reverse SSH?

- Your work or home machine runs **ssh server**
- However, it’s behind NAT, so you cannot directly connect to it
- But you have some machine with public IP address (like **lnx.cs.smu.ca**)
- You can connect the **device behind NAT** to that machine with public IP and forward the device’s **ssh port** (22) to some port on that public machine
- Now you can ssh to your home device from anywhere just by logging in to that public server and ssh to the port defined on the previous step

Reverse SSH: just a remote port forwarding

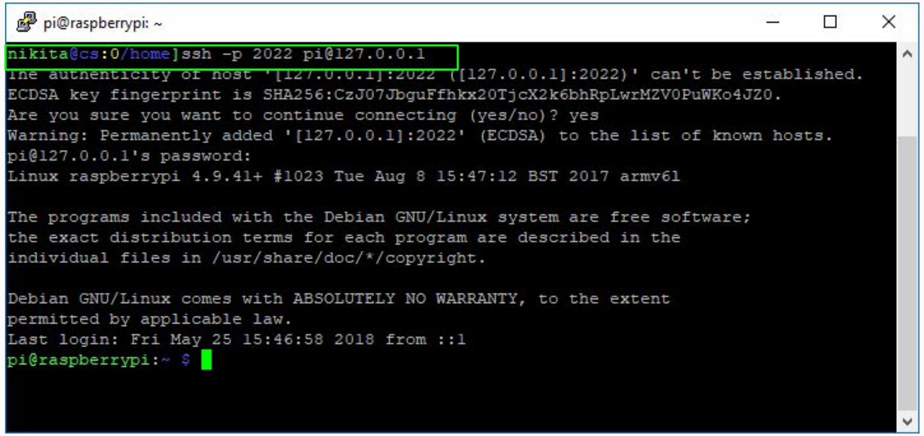
ssh -f -N -R 2022:localhost:22 nikita@cs.smu.ca

Will “map” **localhost:22** (ssh port) from a machine behind NAT to port 2202 at cs.smu.ca



pi@raspberrypi: ~

```
pi@raspberrypi:~ $ ssh -f -N -R 2022:localhost:22 nikita@cs.smu.ca
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.100  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::5f7c:b5ad:b8bd:487e  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:97:28:bf  txqueuelen 1000  (Ethernet)
        RX packets 1784540  bytes 350106862 (333.8 MiB)
        RX errors 0  dropped 69  overruns 0  frame 0
        TX packets 2065372  bytes 358584847 (341.9 MiB)
        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 1  (Local Loopback)
        RX packets 2753  bytes 429224 (419.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2753  bytes 429224 (419.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



pi@raspberrypi: ~

```
nikita@cs:0/home]ssh -p 2022 pi@127.0.0.1
The authenticity of host '[127.0.0.1]:2022 ([127.0.0.1]:2022)' can't be established.
ECDSA key fingerprint is SHA256:CzJ07JbguFfhkx20TjcX2k6bhRpLwrMZV0PuWKo4JZ0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[127.0.0.1]:2022' (ECDSA) to the list of known hosts.
pi@127.0.0.1's password:
Linux raspberrypi 4.9.41+ #1023 Tue Aug 8 15:47:12 BST 2017 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 25 15:46:58 2018 from ::1
pi@raspberrypi:~ $
```

Home machine behind NAT

Machine with public IP address

Exercise

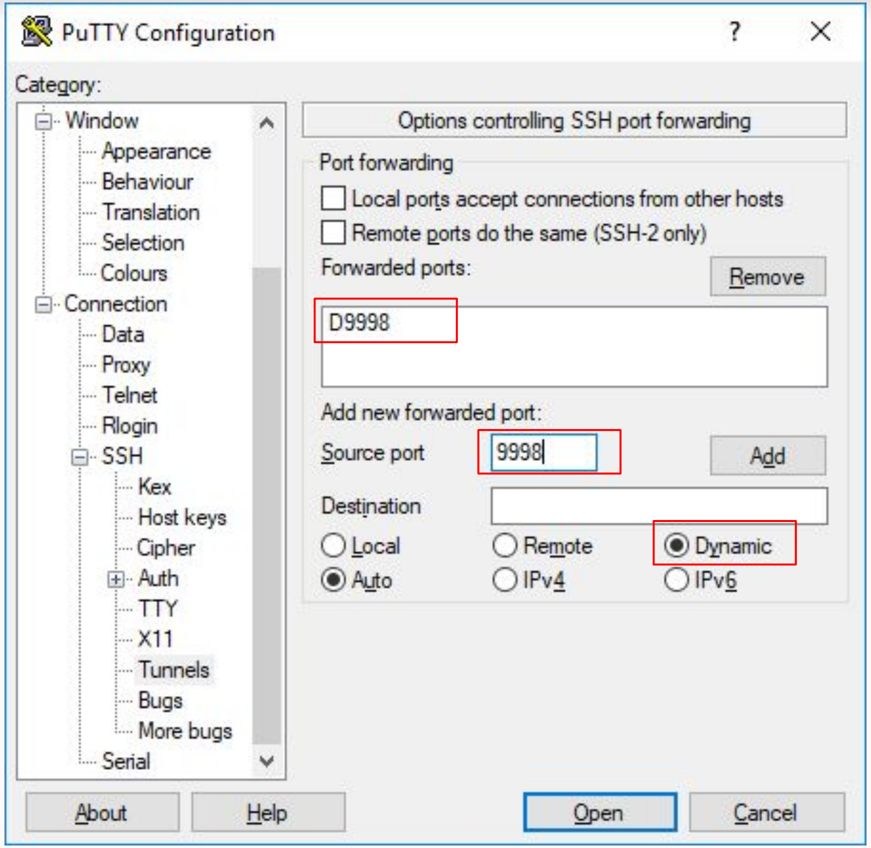
Set up reverse ssh at ***lnx.cs.smu.ca*** *using* ***localhost:port (port = 1MMDD where MM-your month of birth, DD-day of birth)***

- *If you have a virtual machine on your PC, set up reverse ssh to your VM;*
- *Otherwise, set up reverse ssh with dev.cs.smu.ca (just to understand how it works)*

Dynamic Port Forwarding

Dynamic port forwarding turns your **SSH client** into a SOCKS proxy server. SOCKS is a protocol for programs to request any Internet connection through a proxy server. Each program that uses the proxy server needs to be configured specifically, and reconfigured when you stop using the proxy server [l]

Dynamic Port Forwarding: Windows



Options controlling SSH port forwarding

Port forwarding

- [ ] Local ports accept connections from other hosts
- [ ] Remote ports do the same (SSH-2 only)

Forwarded ports: D9998

Add new forwarded port:

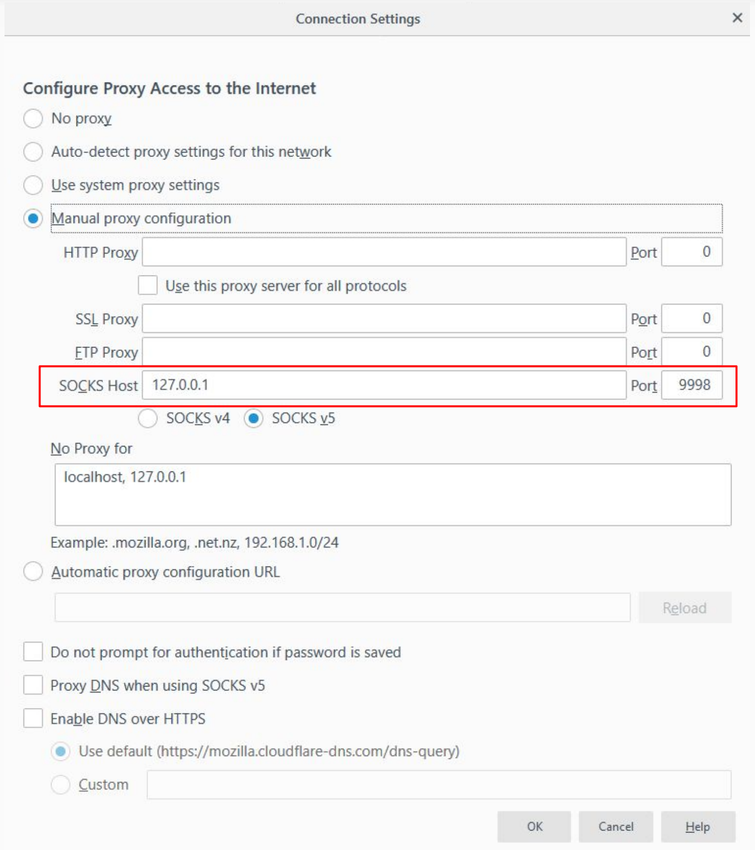
Source port: 9998

Destination:

Local [✓ Dynamic] Remote

[✓ Auto] IPv4 IPv6

Dynamic Port Forwarding: Windows: Firefox



Connection Settings

Configure Proxy Access to the Internet

- ○ No proxy
- ○ Auto-detect proxy settings for this network
- ○ Use system proxy settings
- ● Manual proxy configuration

HTTP Proxy		Port	0
	☐ Use this proxy server for all protocols		
SSL Proxy		Port	0
FTP Proxy		Port	0
SOCKS Host	127.0.0.1	Port	9998

○ SOCKS v4 ● SOCKS v5

No Proxy for

localhost, 127.0.0.1

Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL

Reload

- ☐ Do not prompt for authentication if password is saved
- ☐ Proxy DNS when using SOCKS v5
- ☐ Enable DNS over HTTPS
  - ● Use default (https://mozilla.cloudflare-dns.com/dns-query)
  - ○ Custom

Dynamic Port Forwarding: Linux

```
ssh -f -N -C -D 9998 username@dev.cs.smu.ca
```

- -C Compress data
- -D Dynamic port forwarding