SSH client

- Simplest form:

```
$ ssh hostname
```

- Different username on server:

```
$ ssh user@hostname
```

- Execute remote command:

```
$ ssh user@hostname command
```

- General form:

```
$ ssh <options> <option1+parameter1>
<optionN+parameterN> <user@>hostname
command
```

- Authentication:
  - host-based
  - public key
  - challenge-response
  - password

SSH keys

- Generate keys:

```
$ ssh-keygen -t dsa
```

- Usage:

SSH client

user1:
~/.ssh/public_key1
~/.ssh/private_key1

~/.ssh/authorized_keys:

public_key2

SSH server

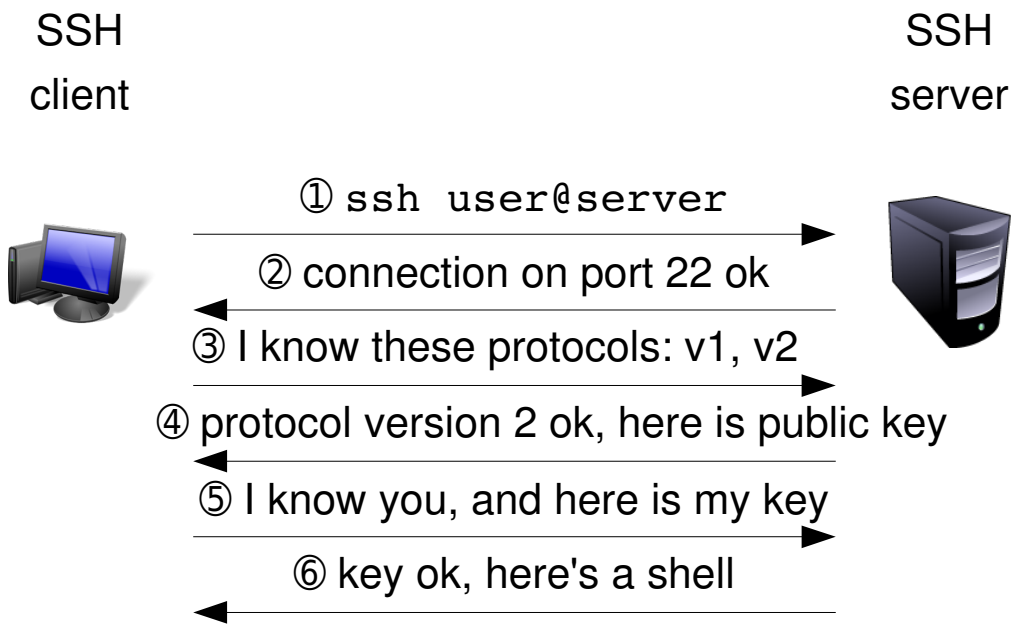
user2
~/.ssh/public_key2
~/.ssh/private_key2

~/.ssh/authorized_keys:

public_key1

SSH client configuration

- A typical SSH connection:



- Check/debug connection:

```
$ ssh -v user@server
```

- Increase verbosity:

```
$ ssh -vv user@server
$ ssh -vvv user@server
```

Secure copy and secure FTP

- Example `/etc/ssh/ssh_config`:

```
Host *
    HashKnownHosts yes
    GSSAPIAuthentication yes
    GSSAPIDelegateCredentials no
```

- Example `~/.ssh/config`:

```
Host atelier.atcomputing.nl
  User petra
  NoHostAuthenticationForLocalhost yes
Host ny.atcomputing.nl
  SendEnv LANG LC_*
Host *
  StrictHostKeyChecking yes
```

- Syntax `scp`:

```
$ scp <options> <option1 parameter1>
<optionN parameterN>
<<user1@>host1:>/path/to/file1
<<user2@>host2:>/path/to/file2
```

- Syntax `sftp`:

```
$ sftp <options> <option1 parameter1>
<optionN parameterN> <user@>host
```

SSH agent

- Start agent:

```
$ ssh-agent
```

- Add keys:

```
$ ssh-add <options> <file>
```

Options:

- `-c`: ask for approval before usage
- `-D`: remove all keys from agent
- `-d`: remove specific key from agent
- `-e reader`: remove key from smartcardreader
- `-L`: list public keys
- `-l`: list public key fingerprints
- `-s reader`: add key to smartcardreader
- `-t life`: validity of a key
- `-X`: unlock agent
- `-x`: lock agent with password

SSH agent forwarding

- Via the command line:

```
$ ssh -A user@host
```

- Via configuration file:
  - Do not allow global forwarding:

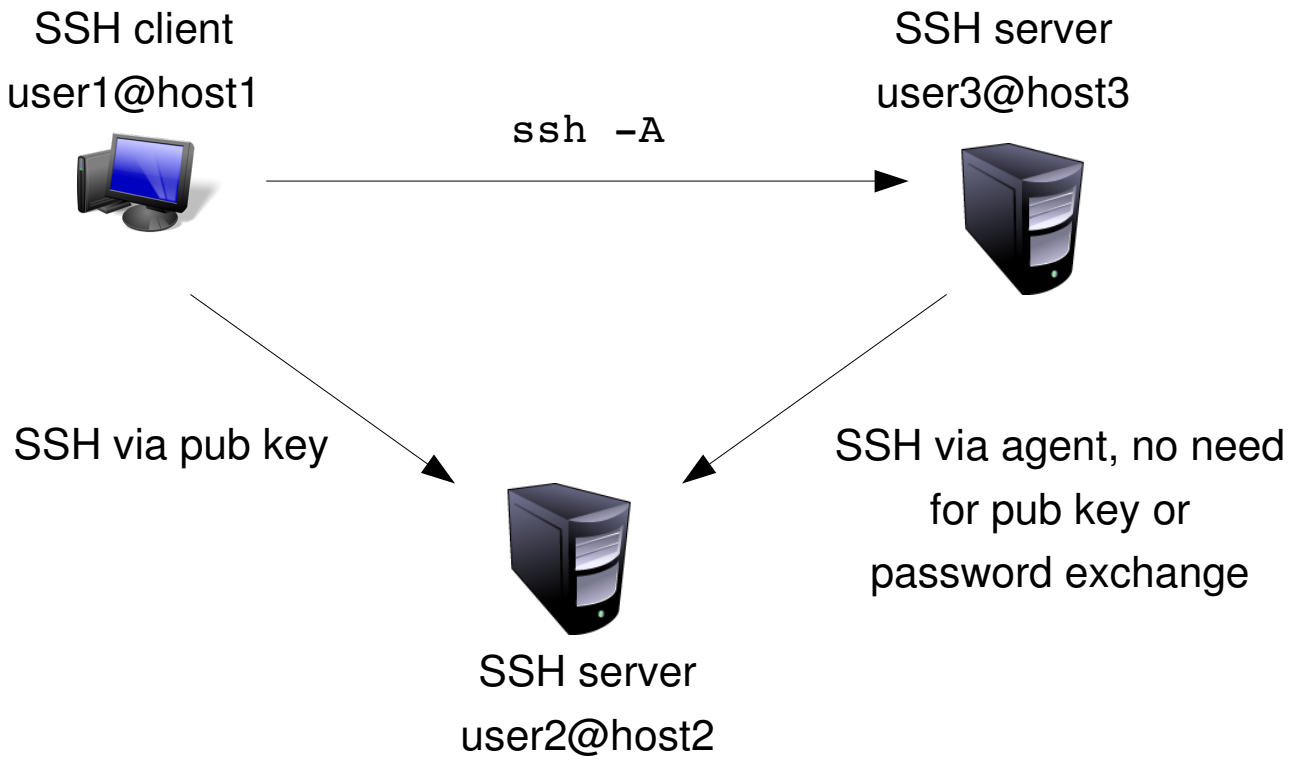
    `/etc/ssh/ssh_config`:

```
ForwardAgent no
```

  - Per host in user config: `~/.ssh/config`:

```
Host trusted.atcomputing.nl
  ForwardAgent yes
Host *
  ForwardAgent no
```

How it works:



SSH daemon configuratie

- Example `/etc/ssh/sshd_config`:

```
Port 22
Protocol 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
UsePrivilegeSeparation yes
SyslogFacility AUTH
LogLevel INFO
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
PubkeyAuthentication yes
HostbasedAuthentication no
ChallengeResponseAuthentication no
X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
AcceptEnv LANG LC_*
Subsystem sftp /usr/lib/openssh/sftp-server
UsePAM yes
```

- Test config:

```
# /usr/sbin/sshd -t
```

(requires full path to SSH daemon)

SSH processes

- Server administration:

```
# /etc/init.d/sshd [ stop | start | reload | restart ]
```

- or:

```
# service sshd [ stop | start | reload | restart ]
```

- Start manually:

```
# sshd <options> <option value>
```

- Monitor daemon:

```
# sshd -D
```

- Test config file:

```
# sshd -t /path/to/new/sshd_config
```

- Only log fatal errors:

```
# sshd -q
```

- Tip for remote configuration:

Make sure you are logged in. Test changes using a second login session. Only reload, don't restart. That way you can repair any errors that might occur.

SSH tunnels

- Syntax:

```
$ ssh -Llocal_port:localhost:remoteport\
user@server
```

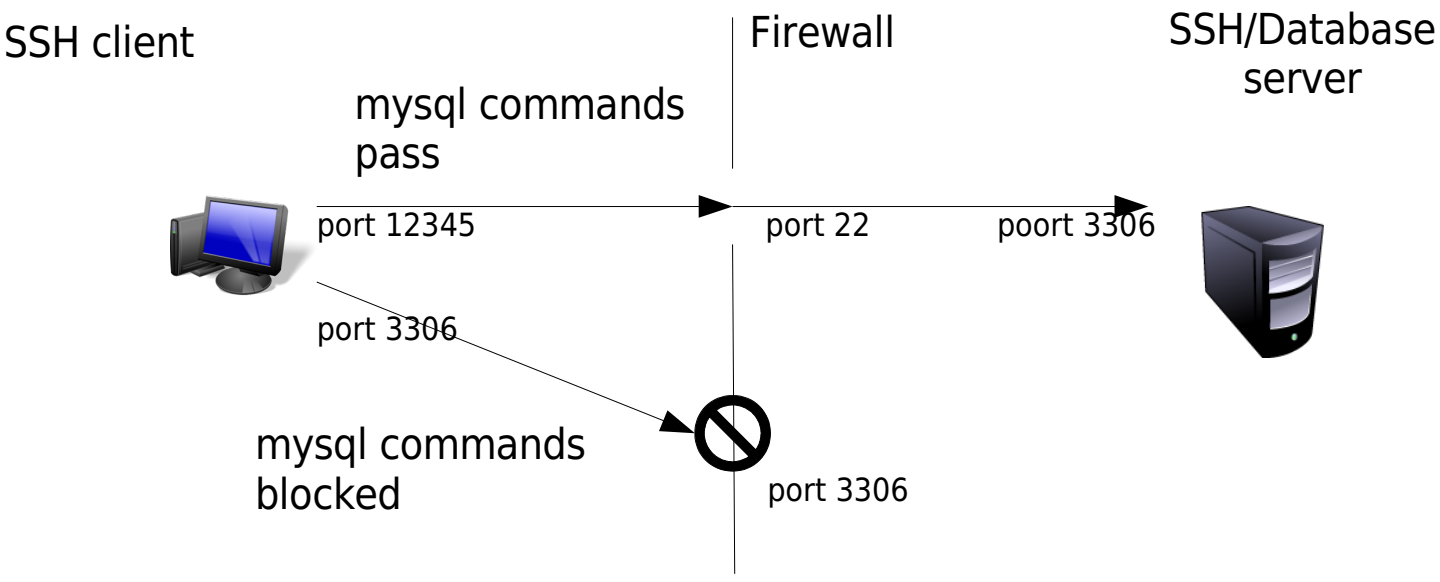
- Options:
  - `-L`: connect local port N1 to port N2 on the SSH server
- Example of a tunnel setup:

```
$ ssh -l mysqladmin -L12345:127.0.0.1:3306\
databaseserver
```

- Example of the usage of a tunnel:

```
$ mysql -u admin -p -h 127.0.0.1 -P 12345
```

How it works:



X11 forwarding

- Activate X11 forwarding:

```
$ ssh -X servername
```

- or:

```
$ ssh -o “ForwardX11=yes” servername
```

- Or in `/etc/ssh/ssh_config` or `~/.ssh/sshdconfig`:

```
ForwardX11 yes
```

- On the server in `/etc/ssh/sshd_config`:

```
X11Forwarding yes
X11DisplayOffset 10
```

- Test config:

```
$ ssh server
user@server's password:
$ echo $DISPLAY
:10.0
$ xclock &
```