SFTP account configuration
Summary

Introduction

SFTP account configuration
  Step 1: Creating an SSH key
  Step 2: Gathering your information
  Step 3: Communicate your identifiers
  Step 4: Configure your client tool

Key authentication
  Operation principle
  Security principle
  Recommendations
  Creating an SSH key on Windows
  Creating an SSH key on Linux or Mac
  Public key format

Configuring an SFTP Client (Filezilla)

Example of an automation script
Introduction

PeopleDoc provides a transit space for different mass actions on the platforms. You can access this space through a secure SFTP connection.

The purpose of this document is to describe the different steps required to set up a SFTP account and the means to connect to it.

SFTP account configuration

- **Step 1: Creating an SSH key**

An SSH key (Secure Shell key) allows you to authenticate securely.
  - If you are on Windows: [Creating an SSH key on Windows](#)
  - If you are running Linux or Mac: [Creating an SSH key on Linux or Mac](#)

- **Step 2: Gathering your information**

In order to create an SFTP account, we need the following information:
  - Your public IP addresses (workstation and/or server)
  - Your public key

Check with your *IT department* to find out which public IP addresses will be used.

- **Step 3: Communicate your identifiers**

Once authorized on our server, PeopleDoc gives you the necessary information to connect to your account.

These informations are:
  - Your username
  - The IP address of the SFTP server
  - The server port

Check with your *IT department* to make sure you can access the server at the requested IP address and port.
• **Step 4: Configure your client tool**

Once all the informations are transmitted, you can configure your client to log in.

- Example configuration with FileZilla: [Configuring an SFTP Client (Filezilla)]
- Python Automation Script Example: [Example of an automation script]

You can also find the connection information for PeopleDoc SFTP servers in the documentation available at this address:
**Key authentication**

- **Operation principle**

  Key authentication works through 3 components:
  
  - A private key: It allows to prove its identity to the SFTP server.
  - A passphrase (optional): It allows to secure the private key.
  - Public key: allows the server to authorize the corresponding private key.

  The public key must be provided to PeopleDoc to configure your access.

- **Security principle**

  - The SFTP service of PeopleDoc accepts SSH keys of the type "**ED25519**" or "**RSA**" whose size is **greater than or equal to 2048 bits**.
  - SSH keys of type "**DSA**, "**ECDSA**" or any RSA key less than 2048 bits do not allow access to the SFTP service of PeopleDoc.

- **Recommendations**

  - Give a comment to each of your SSH keys to differentiate them, which makes it easier to communicate with multiple public key permissions.
  - Assign a passphrase to your SSH key during generation to ensure its security. This passphrase will only be known to you alone.

For more information about security related to using SSH keys:
Creating an SSH key on Windows

Download and install **PuttyGen** which will allow you to generate your public key and your private key.

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

At the launch of **PuttyGen**, you arrive on the following screen:

Select settings:
- **Type of key to generate**: ED25519 or RSA
- **RSA Number of bits in a generated key**: 2048 minimum (4096 recommended)

Then click on **Generate**.
To generate keys, you must move your mouse pointer in the active area of the window: the utility will use these movements as a source of random data.
After generating the keys, assign a comment to identify your public key.

- **Key comment**: Identification of the owner of the key (date + email address)
- **Example**: 2019-04-12_Martin_Dupont@Entreprise.fr

It is recommended to secure your private key via passphrase

- **Key passphrase**: Your passphrase

You will notice that the first box now contains your public key in **OpenSSH** format. This one will have to be transmitted to us.

Save the **public key** via the **Save public key** button.

By convention, name it according to the type of key and its owner with the extension ".pub".

- **Save public Key**: id_"key_type"_"owner".pub
- **Example**: id_rsa_Martin_Dupont.pub
The latter will be saved in SSH2 format; Format that can be converted to OpenSSH format. It should be noted that you can re-generate your public key at any time by loading the private key via the Load button.

Then save the private key via the Save private key button. By convention, name it according to the type of key and its owner.

- Save private Key : id_"key_type"_"owner"
- Example : id_rsa_Martin_Dupont

This one must not be sent to us. It will have to be kept and installed on your workstations/servers requiring to connect to our server.
Creating an SSH key on Linux or Mac

In a terminal,

```
ssh-keygen -t KEY_TYPE -b BITS -C "COMMENT" -f "LOCATION & NAME"
```

- **KEY_TYPE**: ed25519 ou rsa
- **RSA BITS**: 2048 minimum (4096 recommended)
- **COMMENT**: Identification of the owner of the key (date + email address)
- **LOCATION**: `/home/your_user/.ssh/id_"key_type"_"owner"

**Example:**

```
ssh-keygen -t rsa -b 8192 -C "2019-04-12_email_user@enterprise" -f "/home/user/.ssh/id_rsa_user"

ssh-keygen -t ed25519 -C "2019-04-12_Martin_Dupont@Entreprise" -f "/home/Martin_Dupont/.ssh/id_rsa_Martin_Dupont"
```

When executing the command, a passphrase will be asked.

It is recommended to secure your private key via passphrase.

The command generated two files.

- your private key  
  `
  (id_rsa_Martin_Dupont)
  `
- your public key  
  `
  (id_rsa_Martin_Dupont.pub)
  `

```
-rw------- 1 Martin_Dupont Martin_Dupont 3243 avril 12 12:16 id_rsa_Martin_Dupont
-rw-r--r-- 1 Martin_Dupont Martin_Dupont 761 avril 12 12:16 id_rsa_Martin_Dupont.pub
```

Be careful never to alter the rights on these files at the risk of not being able to authenticate you.
The public key (ending with ".pub") should be transmitted to us. The private key should not be sent to us. It will have to be kept and installed on your workstations/servers requiring to connect to our server.

- Public key format

The public key generated is in OpenSSH format. This format is composed of 3 parts:

- The type of the key: ssh-rsa
- The key: AAAA ..... KQ==
- The comment: 2019-04-12_Martin_Dupont@Entreprise.fr

```
cat id_rsa_Martin_Dupont.pub

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAAT+eQkC8dUytS5Yl5mzJ/nf32hVczxMW0zmRkl9Yr+DO4W
2nuIvSeHk9Y0xWf2X4RzZOAjRzU1pPv7wJ9kyljJ5qR9a6qLXvj15m8v9nsgnF4yYsTjW
+9nsDjJ6z3kg+e14t+Jx00sD11t0Y4SfLZ6aQz6W/56Gr2f/9+GjZuJrX57p98JQp7X64
oHr9NKi2Lg8xM39TJj3wcE/RW+2f4x009ZsK+3Yf7syw5x7NnA8c5s4/P4mXe+wT
89LTVH3F04aDw7m5UOeqdOablqzTvWq4s0Gv0yA/FJ5C/Dn9/TbOlHn/CNfH7W
1OvxWn/lRv28yPIM89b0x55w8bE2+3u2/nEhB9c0J7x1/miVD
== 2019-04-12_Martin_Dupont@Entreprise.fr
```

Only the public key is to be sent to PeopleDoc. Be careful never to alter the contents of this file when it is transferred.
Configuring an SFTP Client (Filezilla)

Beforehand, you must have provided PeopleDoc with the following information:
- Your public IP addresses (workstation and/or server)
- Your public key

PeopleDoc must send you back:
- Your login
- The server address
- The server port

Download and install FileZilla: https://filezilla-project.org/download.php?type=client

Click on File -> Site Manager -> New Site.
Fill in the fields according to the server you want to connect to.

<table>
<thead>
<tr>
<th>Protocole</th>
<th>SFTP</th>
</tr>
</thead>
</table>
| Host      | SFTP server address (DNS)  
example : adresse_sftp_server.people-doc.com |
| Port      | 9030 |
| Logon Type| Key file |
| User      | Your login transmitted by PeopleDoc |
| Key file  | Path to your private key  
example : c:\users\Martin_Dupont\id_rsa_Martin_Dupont |

Finally, click on Connect (the profile of the Site will be automatically saved)

If a passphrase has been defined when you create your key pair (public and private), FileZilla will prompt you to fill it when you connect.

When you first sign in, a warning appears to make sure you are connecting to the desired server.
If the fingerprint is the same as the server you want to access (RSA key fingerprint or SSH public key (PEM format) from the server on the official documentation), you can trust this server and associate it with the connection.
If not, please let us know.
**Example of an automation script**

For production, it is highly recommended to automate the transfer of files from your servers to our SFTP server.


```python
#!/usr/bin/env python
import paramiko
paramiko.util.log_to_file('/tmp/paramiko.log')

# Connection
host = 'address_sftp_server.people-doc.com'
port = 9030
transport = paramiko.Transport((host, port))

# Authentication
username = 'xxxx'
key_path = '/home/xxx/.ssh/id_rsa'
key_pass = ''
my_key = paramiko.RSAKey.from_private_key_file(key_path, key_pass)
transport.connect(username=username, pkey=my_key)

# SFTP client
sftp = paramiko.SFTPClient.from_transport(transport)

# Upload using .filepart extension to prevent remote processing during the transfer
local_path = '/home/xxxx/ndmat_yyyy_zzzz_sal_20150909.csv'
remote_path = 'in/sal/ndmat_yyyy_zzzz_sal_20150909.csv'
sftp.put(local_path, remote_path + '.filepart')
sftp.rename(remote_path + '.filepart', remote_path)

# Stop gracefully
sftp.close()
transport.close()
```