

Допустим, вам необходимо настроить беспарольный вход по **ssh** (**scp** и **sftp** тоже) на удаленный сервер **host** под пользователем **user**. Если имя Вашего локального пользователя совпадает с удаленным, то **user@** везде можно опустить.

- Создаем открытый и закрытый ключ нашей локальной системы

```
ssh-keygen -t rsa
```

нажимаем «Enter», отказываясь от ключевой фразы.

- если в системе есть программа **ssh-copy-id**, то настраиваем удаленную систему на то, чтобы оно авторизировало **ssh** по открытому ключу

```
ssh-copy-id -i ~/.ssh/id_rsa user@host
```

если порт не стандартный (отличный от 22)

```
ssh-copy-id '-p 2202 -i ~/.ssh/id_rsa user@host'
```

если **ssh-copy-id** нет, то можно сделать это вручную.

Последовательность действий:

- копируем открытый ключ на удаленную систему

```
scp ~/.ssh/id_rsa.pub user@host:~
```

- Авторизируемся на удаленном сервере

```
ssh user@host
```

- заносим открытый ключ нашей локальной системы в авторизованные ключи удаленной системы, устанавливаем правильные права и «убираем за собой мусор»:

```
remote$ [ -d ~/.ssh ] || (mkdir ~/.ssh; chmod 711 ~/.ssh) # создаем
директорию и даём права
remote$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys          # добавляем
открытый ключ
remote$ chmod 600 ~/.ssh/authorized_keys                    # делаем правильные
права
remote$ rm ~/id_rsa.pub                                     # удаляем не нужное
```

- проверяем, что все работает, запускаем на локальном компьютере.

```
ssh user@host
```

From:  
<https://docs.infomir.com.ua/> -

Permanent link:  
[https://docs.infomir.com.ua/doku.php?id=knowledge\\_base:ssh\\_key\\_authorization](https://docs.infomir.com.ua/doku.php?id=knowledge_base:ssh_key_authorization)

Last update: 2021/12/15 14:31



