SSH Client and Server

To login to any machine with SSH, you need to install the ssh server. The client is installed on Debian Wheezy by default. This guide sets up SSH access for BackupPC, but you can use it for any other user respectively. You can find further information in the BackupPC FAQ: SSH Setup.

Important!

After creating the authorized_keys file with the public keys of root@myserver and backuppc@myserver, ssh to the client as root and as backuppc. User backuppc will not be able to establish a connection, but the client's key will be added to known_hosts. This is necessary for BackupPC to work correctly.

SSH Server

```
apt-get install ssh
```

Running ssh-keygen as root will install the host keys, ssh-keygen as <user> the keypair into directory ~/.ssh. The private key file, e.g. ~/.ssh/id_rsa needs to be copied to all clients which need access to the server, the public key file, e.g. ~/.ssh/id_rsa.pub, needs to be added to the authorized_keys file of the server to allow access to the server. In addition, you should adjust the access rights for the different files and directories. As <user>, enter the following commands

On the client:

```
ssh-keygen
ssh -vvv -p <port#> <servername> // check the ssh connection
sftp -oPort=<port#> <servername> // transfer the public key file
sftp > put client_id_rsa.pub
```

On the server:

```
sudo cat client_id_rsa.pub >> ~/.ssh/authorized_keys
```

Make sure permissions and ownership are set correctly:

```
sudo chmod 700 ~/.ssh
sudo chmod 600 ~/.ssh/*
sudo chown -R user:user ~/.ssh
```

Check the ssh access:

```
ssh <servername> whoami // needs to return the username under which the ssh
access was established, e.g. user1
ssh -l root <servername> whoami // needs to return "root", as this
```

```
establishes the ssh access as root, not as user1
ssh -vvv -p <port#> -l root <servername> whoami // same as above, but use
different port number with full debug verbose output
```

Important

- generate the keypair under user rights, not root, e.g. user1
- establish ssh access under user rights by accessing that user's .ssh directory on the server (use the user1 public key)
- establish ssh access under root rights by accessing root's .ssh directory on the server (use the user1 public key)

You might need to include the user in sshd.conf to authorize access.

PuTTY as client

When you are working with private (and public) keys generated by OpenSSH, you will have files called id_rsa and id_rsa.pub. These files can't be used in PuTTY directly. Instead they need to be converted to something else using PuTTYgen, also available from the Putty page.

- Download PuTTYgen.
- Load your key, mine is called id_dsa. Enter your passphrase.
- Save the private key, I saved mine as id_rsa.ppk.

Host keys

How to reset host authentication key for known_hosts

If you receive **RSA host key for foo.bar has changed and you have requested strict checking** , do the following:

- Line **Offending RSA key in ~/.ssh/known_hosts:11** indicates that line 11 contains the violating key, open known_hosts in an editor and delete line 11
- Alternatively you can remove the relevant key by doing the following **ssh-keygen -R 127.0.0.1** (Obviously replace with the servers IP)
- Check the actual fingerprint of the server with **ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub**

If you run Secure Shell app in the Chrome Browser to establish a SSH connection, do the following:

- Open Secure Shell, then open the JavaScript console by typing **Ctrl-Shift-J**
- Enter **term_.command.removeKnownHostByIndex(11)** for the example above, where line **Offending RSA key in /.ssh/known_hosts:11** indicates that line 11 contains the violating key

Aliases

Define an alias for frequently used ssh connections:

```
$ echo "alias <compname> 'ssh -p <port> <compname>'" >> vim ~/.bash_aliases
```