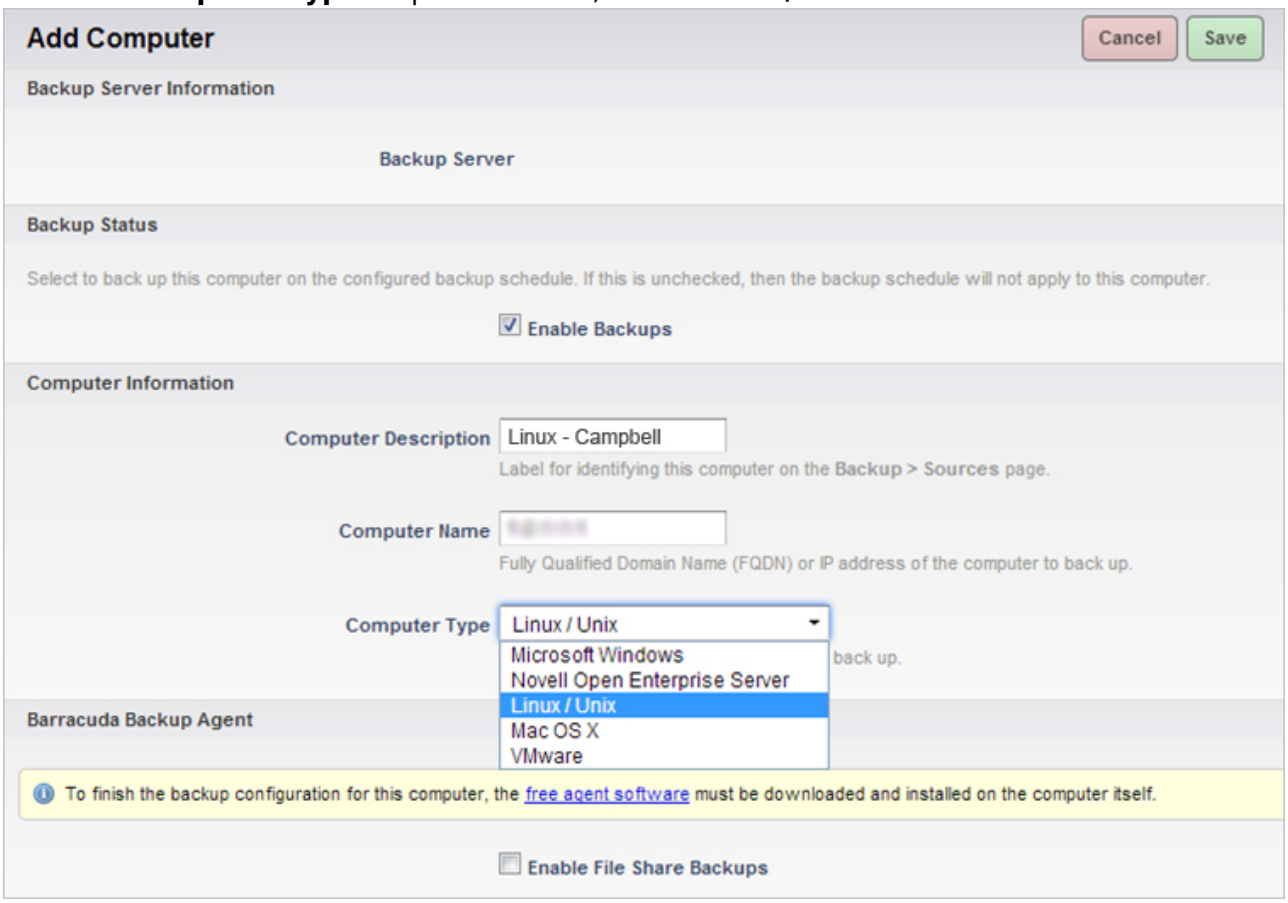
# Network File-Share Backup for Linux/UNIX (SSHFS)

https://campus.barracuda.com/doc/78809322/

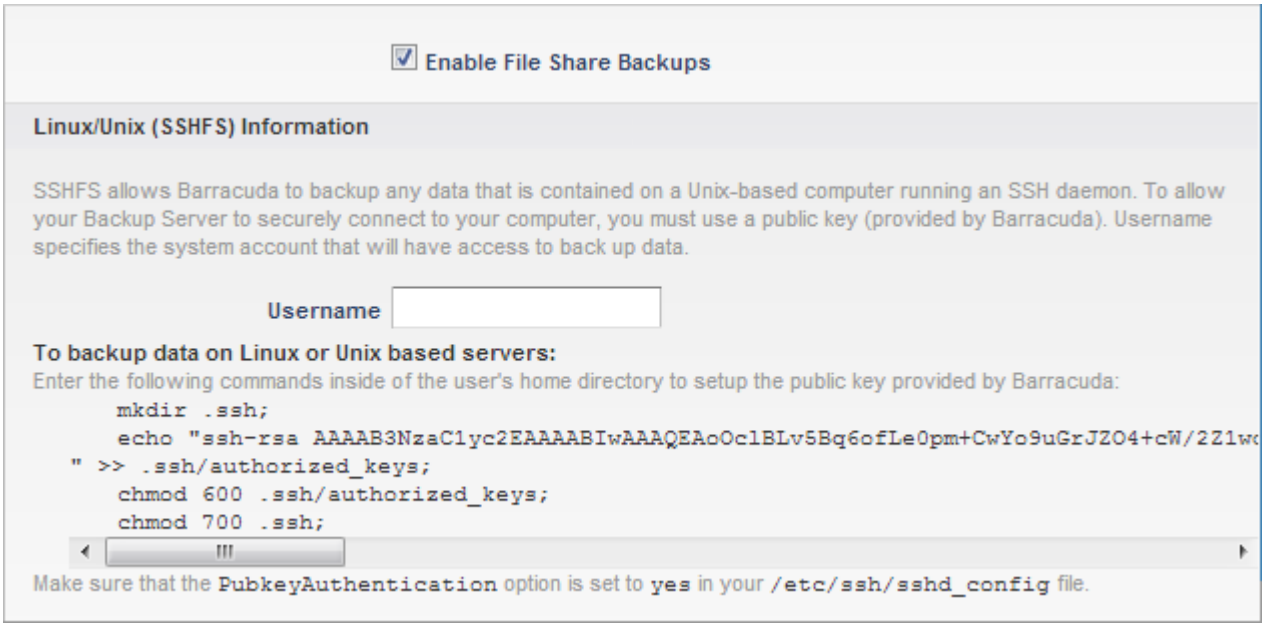
The alternate method of backing up Linux systems–Linux systems the Barracuda Backup Agent for Linux cannot support–and UNIX systems, is the network file-share backup via SSHFS. Prior to configuring the file share backup, ensure that SSH is enabled on the system and that the system account Barracuda Backup is to use has adequate permissions to access the shares, directories, and files it will be protecting.

Use the following steps to configure a file share backup for Linux/UNIX systems using SSHFS:

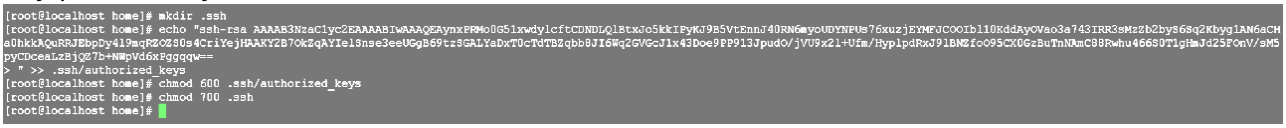
1. Log in to Barracuda Backup and select the associated Barracuda Backup device in the left pane or in the devices table (for customers with multiple Barracuda Backup devices).
2. Go to the **Backup > Sources** page, and click **Add a Computer**.
3. Enter a **computer description** and enter the IP address or fully qualified domain name in the **Computer name** field.
4. From the **Computer Type** drop-down menu, select **Linux / Unix**:




5. Select **Enable File Share Backups**; the **Linux/Unix (SSHFS) Information** section displays:




6. Enter the **Username** for the system account that is to have access to back up data.
7. In the **To backup data on Linux or Unix based servers** section, copy the commands to your clipboard.
8. On the Linux/UNIX system, determine the location of the ssh directory, including the **authorized_keys** file, in the user's home directory:
   - If a *username different than root is used*, place the ssh directory, including the **authorized_keys** file, in the user's home directory.
   - If the *root account is used*, place the ssh directory, including the **authorized_keys** file, in the root home directory
9. Log in to the Linux/UNIX system using an ssh client such as PuTTY, and paste and run the commands copied in *step 7* into the appropriate directory based on *step 8* to set up the public key provided by Barracuda Networks:



```
#RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     .ssh/authorized_keys
#AuthorizedKeysCommand none
#AuthorizedKeysCommandRunAs nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no
```

10. On the Linux/UNIX system, navigate to **/etc/ssh** and open the file **sshd_config** file:

```
[root@localhost mhaag]# vi /etc/ssh/sshd_config
```

    If an error message stating that no **sshd_config** file could be found displays, the ssh daemon may not be installed. In some Linux installations, the ssh client is installed without the ssh daemon.
    For more information, refer to the documentation available online or included with your version of Linux for the proper procedures to install the ssh daemon on your server. For your reference, the following link is provided to show an example of sshd installation instructions, for example, see the Open SSHServer documentation.

11. In the file **sshd_config**, locate the line `PubkeyAuthentication yes`, remove the pound sign "#":

```
#RSAAuthentication yes
ubkeyAuthentication yes
#AuthorizedKeysFile     .ssh/authorized_keys
#AuthorizedKeysCommand none
#AuthorizedKeysCommandRunAs nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no
```

12. Run the following command to restart sshd:
    `/etc/init.d/sshd restart`
    Depending on the Linux Server distro, it may be necessary to disable SELinux enforcement using the command:
    `echo 0 > /selinux/enforce`
13. When the configuration is complete, click **Save** in the Barracuda Backup web interface. The **Add Data Source** page displays. Continue with *Configure Linux/UNIX File Shares for Backup* below.

#### Configure Linux/UNIX File Share for Backup

Now that the Linux/UNIX system is configured for SSHFS backup in the section above, you can begin adding specific shares to back up. You can backup at the root level and back up all child directories or connect to specific shares/directories one-by-one.

To configure Linux/UNIX file share for backup:

1. On the **Add Data Source** page, enter a **Data Description** for this source/share.
2. From the **Data Type** drop-down menu, select **File Share - SSHFS**.
3. In the **File Share Information** section, enter the full path of the directory to be backed up in the **Share Name** field, and click **Test Share**; if the connection is successful, a message displays the connection status, for example, **Status: Successfully connected to computer**. If the connection is not successful, verify you can connect to the share with the configured username, access the system from Barracuda Backup, or that Barracuda Backup has the correct permissions to access the data.
4. If you have successfully added the share, click **Save**. You can continue adding shares by clicking Add **Data Source** for this system on the **Backup > Sources** page, and repeating steps

1 - 3.

**Important**
When creating the backup schedule for a Linux/UNIX system, if you are backing up `root` or "`/`", you must *clear* or create an exclusion rule for the **proc** and **sys** directories.

## Figures

1. add_linux_system.jpg
2. linuxkey.png
3. linuxServerCode_gray.png
4. poundSignGray.png
5. sshd_config_gray.png
6. pound sign_gray.png