C4 M5 L1 Qwiklab: Using BackupPc to Manage Backups

2 hours Free

Rate Lab

Introduction

Managing backups is an activity of critical importance that you'll have to deal with as a system administrator. Managing backups refers not only to the initial setup of certain backup software but also to ensuring that backups work correctly and can be restored reliably.

In this lab, you'll install and configure backup infrastructure for both local and remote machines to manage backups. You'll also verify that backups can be restored successfully.

In order to do this, we will build on many of the skills previously covered in labs related to administering Linux and Windows machines. This lab is a bit more complex than previous labs. But don't panic! We will guide you all along the way.

Heads up: Make sure to click the **"Start Lab"** button at the top of the screen. It may take a while for the lab to load. Please wait until the lab is running. To mark this lab as completed, make sure to click **"End Lab"** when you're done!

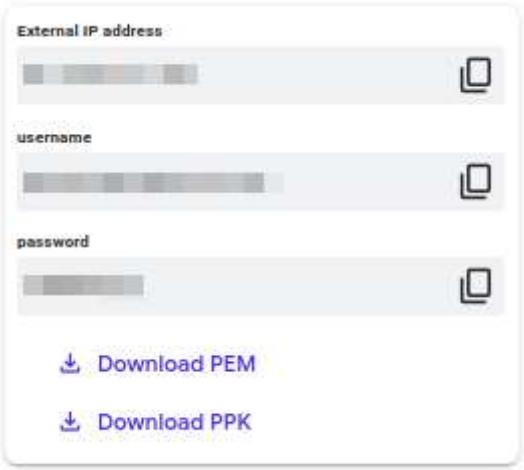
You'll have 120 minutes to complete this lab.

Start the lab

You'll need to start the lab before you can access the materials in the virtual machine OS. To do this, click the green “Start Lab” button at the top of the screen.

Note: For this lab, you are going to access **Linux and Windows VM** through your **local SSH and RDP Client**, and not use the **Google Console** (**Open GCP Console** button is not available for this lab).

After you click the “Start Lab” button, you will see all the connection details on the left-hand side of your screen. You should have a screen that looks like this:



Note: Working with Qwiklabs may be similar to the work you'd perform as an IT Support Specialist; you'll be interfacing with a cutting-edge technology that requires multiple steps to access, and perhaps healthy doses of patience and persistence(!). You'll also be using **RDP** and **SSH** to enter the labs -- critical skills in IT Support that you’ll be able to practice through the labs.

Accessing the linux virtual machine

Please find one of the three relevant options below based on your device's operating system.

Option 1: Windows Users: Connecting to your VM

In this section, you will use the PuTTY Secure Shell (SSH) client and your VM’s External IP address to connect.

Download your PPK key file

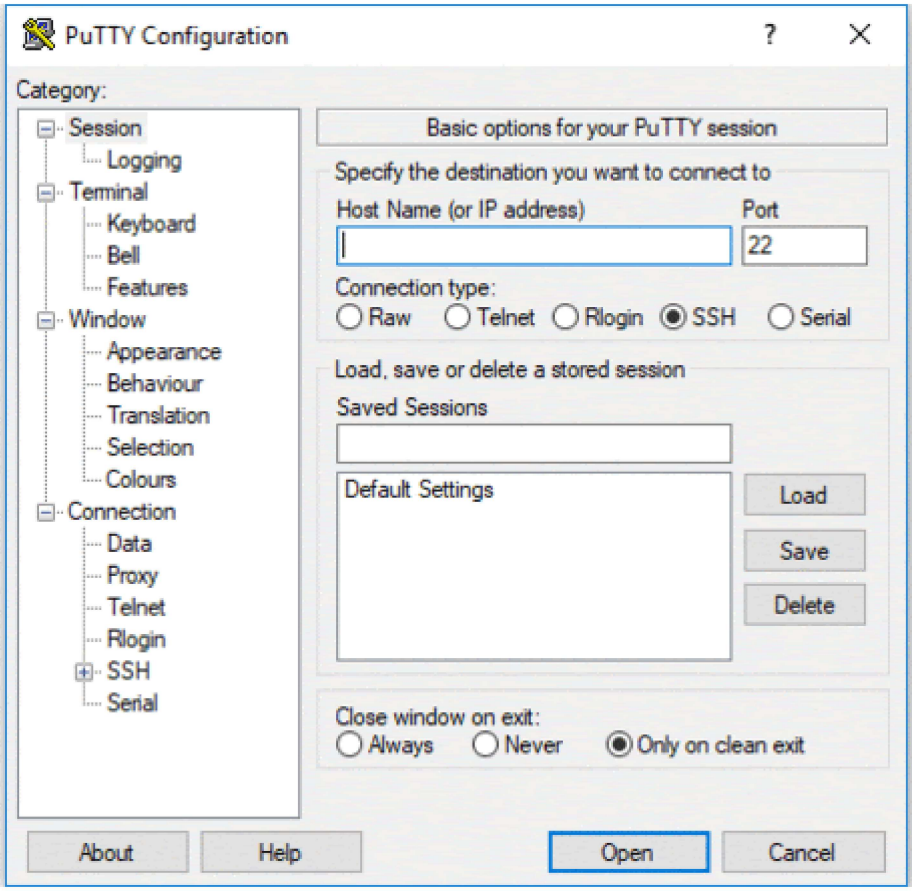
You can download the VM’s private key file in the PuTTY-compatible **PPK** format from the Qwiklabs Start Lab page. Click on **Download PPK**.

Connect to your VM using SSH and PuTTY

1. You can download Putty from here

2. In the **Host Name (or IP address)** box, enter username@external_ip_address.

Note: Replace **username** and **external_ip_address** with values provided in the lab.



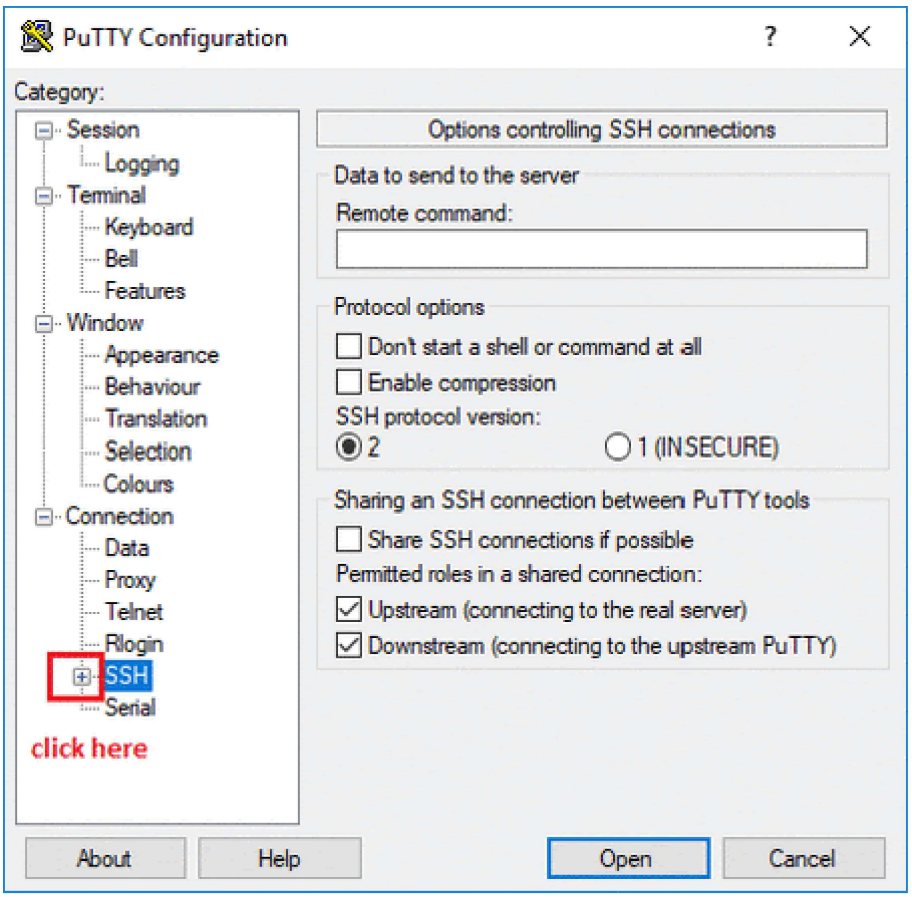
3. In the **Category** list, expand **SSH**.

4. Click **Auth** (don't expand it).

5. In the **Private key file for authentication** box, browse to the PPK file that you downloaded and double-click it.

6. Click on the **Open** button.

Note: PPK file is to be imported into PuTTY tool using the Browse option available in it. It should not be opened directly but only to be used in PuTTY.



7. Click **Yes** when prompted to allow a first connection to this remote SSH server. Because you are using a key pair for authentication, you will not be prompted for a password.

Common issues

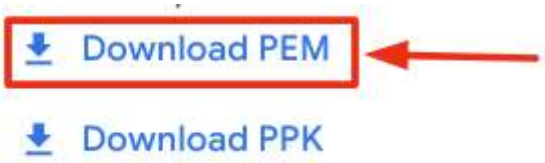
If PuTTY fails to connect to your Linux VM, verify that:

- You entered **<username>@<external ip address>** in PuTTY.
- You downloaded the fresh new PPK file for this lab from Qwiklabs.
- You are using the downloaded PPK file in PuTTY.

Option 2: OSX and Linux users: Connecting to your VM via SSH

Download your VM's private key file.

You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.



Connect to the VM using the local Terminal application

A **terminal** is a program which provides a **text-based interface for typing commands**. Here you will use your terminal as an SSH client to connect with lab provided Linux VM.

1. Open the Terminal application.
    - To open the terminal in Linux use the shortcut key **Ctrl+Alt+t**.
    - To open terminal in **Mac** (OSX) enter **cmd + space** and search for **terminal**.
2. Enter the following commands.

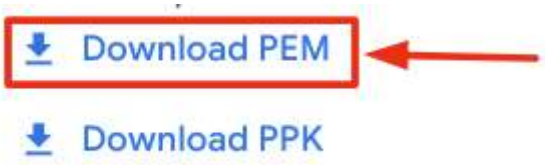
Note: Substitute the **path/filename for the PEM** file you downloaded, **username** and **External IP Address**.
You will most likely find the PEM file in **Downloads**. If you have not changed the download settings of your system, then the path of the PEM key will be **~/Downloads/qwikLABS-XXXXX.pem**

```
chmod 600 ~/Downloads/qwikLABS-XXXXX.pem
ssh -i ~/Downloads/qwikLABS-XXXXX.pem username@External Ip Address
```

Option 3: Chrome OS users: Connecting to your VM via SSH

Note: Make sure you are not in **Incognito/Private mode** while launching the application.
Download your VM's private key file.

You can download the private key file in PEM format from the Qwiklabs Start Lab page. Click on **Download PEM**.



Connect to your VM

1. Add Secure Shell from here to your Chrome browser.
2. Open the Secure Shell app and click on **[New Connection]**.

3. In the **username** section, enter the username given in the Connection Details Panel of the lab. And for the **hostname** section, enter the external IP of your VM instance that is mentioned in the Connection Details Panel of the lab.

4. In the **Identity** section, import the downloaded PEM key by clicking on the **Import…** button beside the field. Choose your PEM key and click on the **OPEN** button.

Note: If the key is still not available after importing it, refresh the application, and select it from the **Identity** drop-down menu.

5. Once your key is uploaded, click on the **[ENTER] Connect** button below.

6. For any prompts, type **yes** to continue.

7. You have now successfully connected to your Linux VM.

You're now ready to continue with the lab!

Linux commands reminder

In this lab, we'll use a number of Linux commands. These were already explained in this and the previous course. This is a reminder of what these commands do:

- `sudo <command>`: executes a command with administrator rights
- `apt update`: updates the list of available packages to be installed
- `apt install package`: installs the given package in the system
- `a2enmod <module>`: enables an Apache2 module
- `a2ensite <module>`: enables an Apache2 website
- `nano <file>`: opens a text editor to edit the file
- `service <service name> restart`: restarts the indicated service
- `ls <directory>`: lists the files in a directory
- `cp <old> <new>`: creates a copy of the old file with the new name
- `cat <file>`: outputs the whole contents of a file
- `grep <pattern> <file>`: filters the text of a file according to the pattern
- `tail <file>`: shows the last lines of a file

Additionally, you can combine these commands using the `|` sign. For example:

```
cat /var/log/syslog | grep error | tail
```

Will first print the output of `/var/log/syslog`, then keep only the lines that say "error" and then keep only the last 10 lines of that output.

We will also present a number of new commands like `htpasswd` or `su`. We will briefly explain what these commands do when they are shown. Remember that you can always read the manual page using `man <command_name>` to learn more about a command.

While you can copy and paste the commands that are presented in this lab, we recommend typing them out manually, to help with understanding and remembering these commands.

Scenario

There are many backup solutions to choose from when backing up your data. Each of these systems has defined goals and situations. In this lab we will be using BackupPc. This solution allows you to backup data across operating systems and has a handy web interface for configuring it.

During the lab, we will configure the machine called **backup-server** to be a server. It will back up data from other machines using BackupPc. We will see how to backup data from the local machine as well as from Linux and Windows remote machines. These are called **linux-instance** and **windows-instance** respectively.

For the backup server, it is important to back up the `/etc` directory where the configurations are stored. For the remote machines, we want to back up the `/home` and `Users` directories where the users' data is stored.

Note: Along the way we will encounter some minor problems that need to be fixed. We will fix them to ensure that backups run smoothly. Encountering these kinds of issues is normal and being able to fix them is a core skill of a system administrator.

Installing and configuring BackupPc

As mentioned, **backup-server** is the machine where the backup infrastructure will run. We need to connect to that machine now.

Connect to **backup-server** using `backup-pc external IP address` from Connection Details Panel and following the instructions given in the section `Accessing the linux virtual machine`. Click on `Accessing the linux virtual machine` from the navigation pane at the right side.

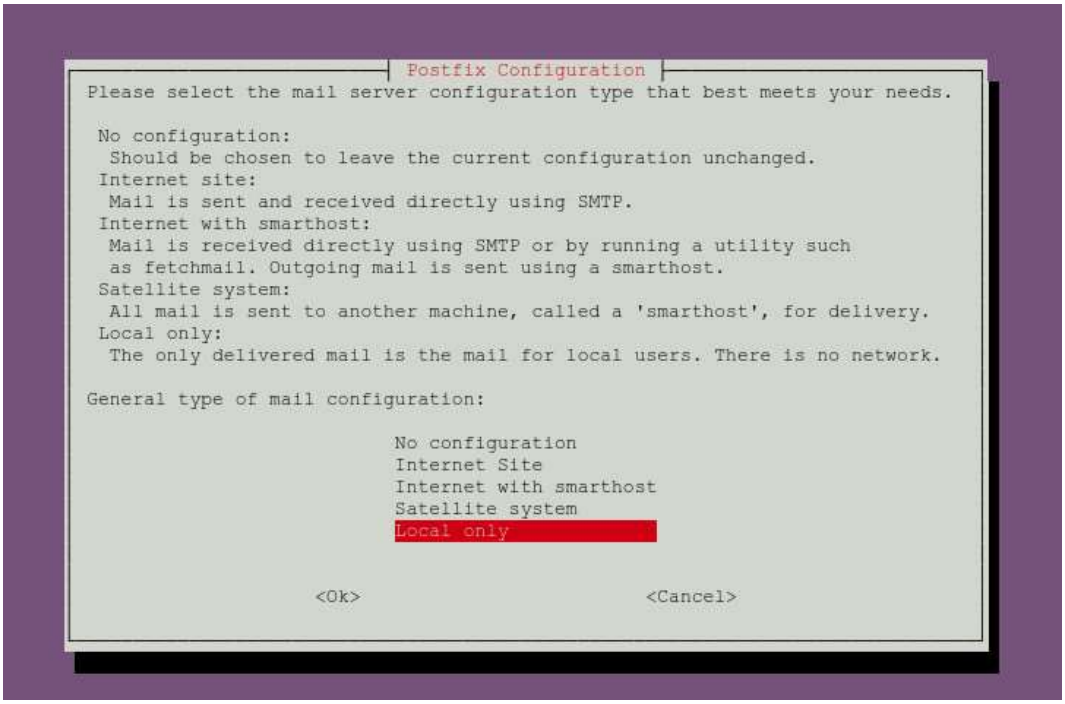
Step 1: Once connected, let's start by installing the `backuppc` package with the following commands:

```
sudo apt update
sudo apt install backuppc
```

```
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils fontconfig fontconfig-config fonts-dejavu-core libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libarchive-zip-perl libarchive13 libauthen-sasl-perl
  libavahi-client3 libavahi-common-data libavahi-common3 libcairo2 libcgi-fast-perl libcgi-pm-perl libcups2
  libdata-dump-perl libdatrie1 libdbi1 libencode-locale-perl libfcgi-perl libfile-listing-perl
  libfile-rsyncp-perl libfont-afm-perl libfontconfig1 libgraphite2-3 libharfbuzz0b libhtml-form-perl
  libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl
  libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-dirent-perl
  libio-html-perl libio-socket-ssl-perl libjansson4 libldb1 liblua5.2-0 liblwp-mediatypes-perl
  liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl
  libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpixman-1-0 libpython-stdlib libpython2.7
  libpython2.7-minimal libpython2.7-stdlib librrd8 libsmbclient libsocket6-perl libtalloc2 libtdb1 libtevent0
  libthai-data libthai0 libtime-parsedate-perl libtimedate-perl libtry-tiny-perl liburi-perl libwbclient0
  libwww-perl libwww-robotrules-perl libxcb-render0 libxcb-shm0 libxrender1 perl-openssl-defaults postfix python
  python-crypto python-ldb python-minimal python-samba python-talloc python-tdb python2.7 python2.7-minimal
  rrdtool samba-common samba-common-bin samba-libs smbclient ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom w3m | www-browser par2 lrzip
  libdigest-hmac-perl libgssapi-perl cups-common libcrypt-ssleay-perl libauthen-ntlm-perl procmail postfix-mysql
  postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb postfix-sqlite sasl2-bin dovecot-common resolvconf
  postfix-cdb mail-reader postfix-doc python-doc python-tk python-crypto-doc python-gpgme python2.7-doc binutils
  binfmt-support librrds-perl heimdal-clients cifs-utils openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils backuppc fontconfig fontconfig-config fonts-dejavu-core libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libarchive-zip-perl libarchive13 libauthen-sasl-perl
  libavahi-client3 libavahi-common-data libavahi-common3 libcairo2 libcgi-fast-perl libcgi-pm-perl libcups2
  libdata-dump-perl libdatrie1 libdbi1 libencode-locale-perl libfcgi-perl libfile-listing-perl
  libfile-rsyncp-perl libfont-afm-perl libfontconfig1 libgraphite2-3 libharfbuzz0b libhtml-form-perl
  libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl
  libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-dirent-perl
  libio-html-perl libio-socket-ssl-perl libjansson4 libldb1 liblua5.2-0 liblwp-mediatypes-perl
  liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl
  libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpixman-1-0 libpython-stdlib libpython2.7
  libpython2.7-minimal libpython2.7-stdlib librrd8 libsmbclient libsocket6-perl libtalloc2 libtdb1 libtevent0
  libthai-data libthai0 libtime-parsedate-perl libtimedate-perl libtry-tiny-perl liburi-perl libwbclient0
  libwww-perl libwww-robotrules-perl libxcb-render0 libxcb-shm0 libxrender1 perl-openssl-defaults postfix python
  python-crypto python-ldb python-minimal python-samba python-talloc python-tdb python2.7 python2.7-minimal
  rrdtool samba-common samba-common-bin samba-libs smbclient ssl-cert
0 upgraded, 98 newly installed, 0 to remove and 1 not upgraded.
Need to get 23.1 MB of archives.
After this operation, 96.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Before pressing "Y" to accept the installation of the `backuppc` package, take a look at the list of packages that will be installed. The list is long because BackupPc relies on functionality provided by other software. You may notice, for example, that this is installing `apache2`, which will be used for hosting the administrative website.

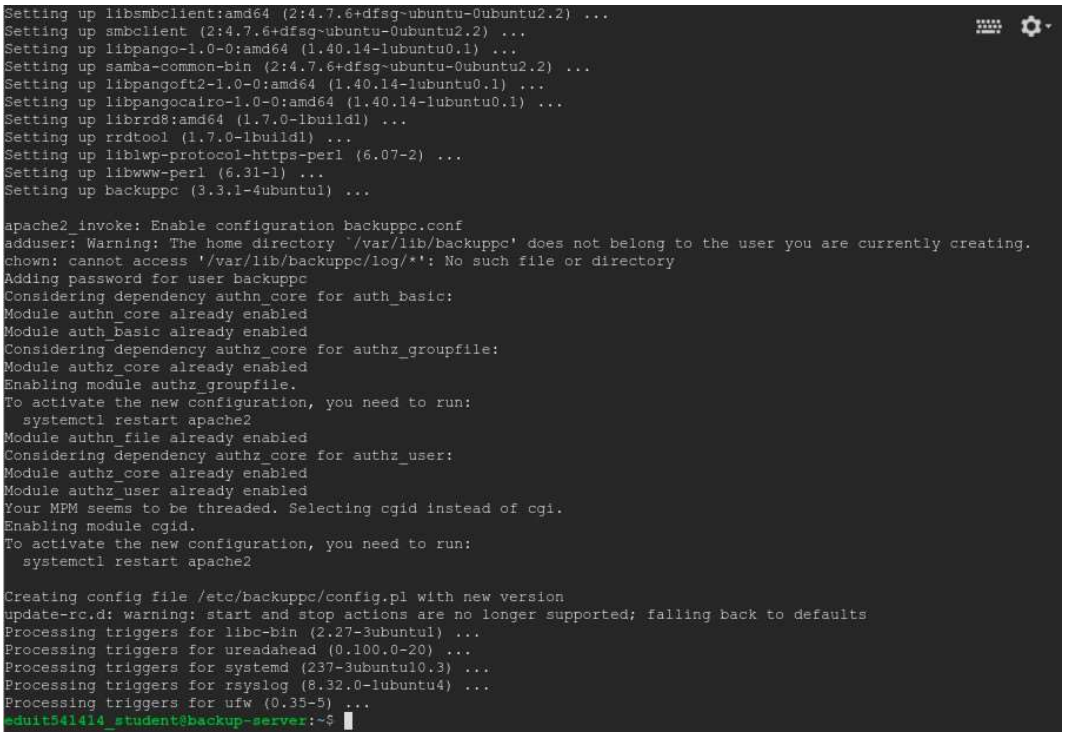
Go ahead and press **Y** now, so that the package and its dependencies can get installed. Once it has downloaded and unpacked the software, it will ask you a number of questions related to the basic configuration of the system.



When the question asks how to configure your email, answer **Local only**.

Then, simply press **ENTER** to accept the pre-selected options for the rest of the questions.

Note: In a real life system, you would configure your backup software to send emails to you if there are any issues. In this case, we don't have a mail system that BackupPc could use to send emails, so we choose to locally store any generated emails.



After confirming the rest of the answers, the package will finish installing. Look at the output and you'll notice that it's telling us that Apache2 needs to be restarted (it suggests using `systemctl restart apache2`, which is an equivalent command to the one we've been using). We will do this in a minute, but first we'll configure Apache2 and BackupPc to use SSL.

Enabling SSL in Apache2

We mentioned before that to encrypt connections, we use HTTPS instead of HTTP. We are going to use the website provided by BackupPc to administer our backups. We want that website encrypted so that the administrative password and any other data entered are not transmitted in plain text. If we didn't do this, anyone eavesdropping on our internet connection could read our admin password and any other data we sent.

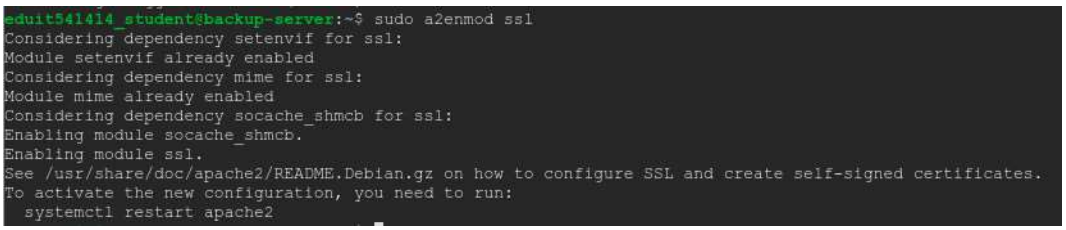
In order to enable HTTPS for our website we need to do three things:

- Enable the SSL module in Apache2.
- Enable the default SSL site in Apache2.
- Require BackupPc to use SSL connections.

Let's get to that.

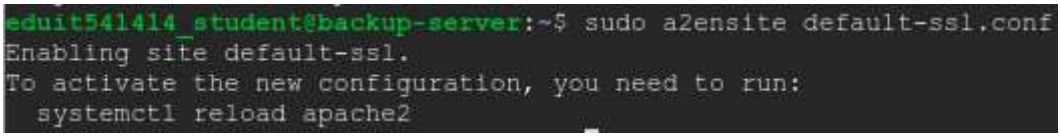
Step 2.1: To enable the SSL module, we'll use the `a2enmod` command.

```
sudo a2enmod ssl
```



Step 2.2: To enable the default SSL site, we'll use the `a2ensite` command.

```
sudo a2ensite default-ssl.conf
```



Enabling SSL in BackupPc and setting the admin password

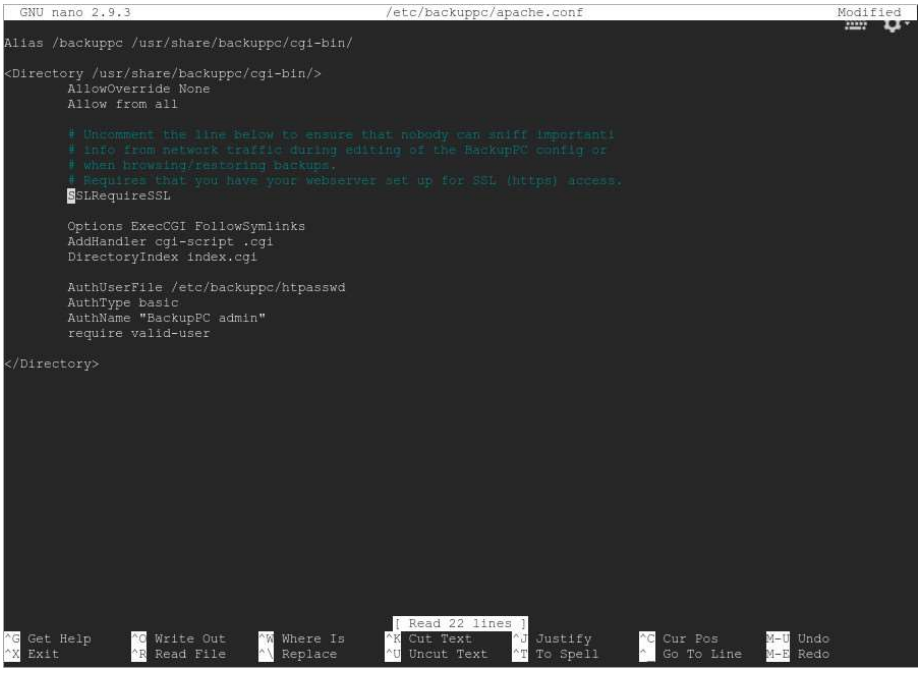
Step 2.3: To require that BackupPc uses SSL connections, we'll edit the configuration file with nano.

```
sudo nano /etc/backuppc/apache.conf
```

This file includes all the settings necessary for BackupPc to have a working website under Apache2. If you want to learn more about the options configured, you can look them up in the Apache2 documentation. For now, let's focus on one specific option: `SSLRequireSSL`.

We will remove the # character in front of the **SSLRequireSSL** option. Which means that we are enabling the option (the "#" character is used in Apache2 config files to indicate lines that will be ignored).

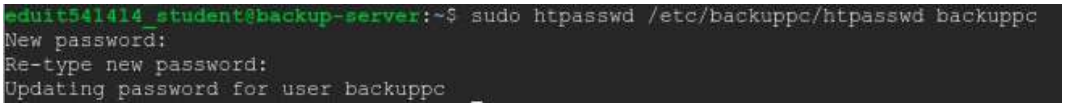
- To edit the file **apache.conf** move your cursor by using down arrow keys present in your system and remove the # character in front of the **SSLRequireSSL** option.



- Once you've removed the character, press **Ctrl-X** to exit, then press **Y** at the prompt to save your changes, and finally press **Enter** at the filename prompt.

Step 3: We are almost done with the initial setup. The only missing step is to set the administrative password of the `backuppc` user. We can do this with the `htpasswd` command, which is used to set passwords for web users.

```
sudo htpasswd /etc/backuppc/htpasswd backuppc
```



Note: Make sure to note down the password on your local editor as it will be used in the following steps.
This command will prompt you for a password, you can enter any password you like, and then enter it again to confirm it. If you forget it, you can repeat this step to set it to a different one.

Step 4: With that, we have our service ready to be used. Let's restart Apache2 now, so that it picks up all of our changes.

```
sudo service apache2 restart
```

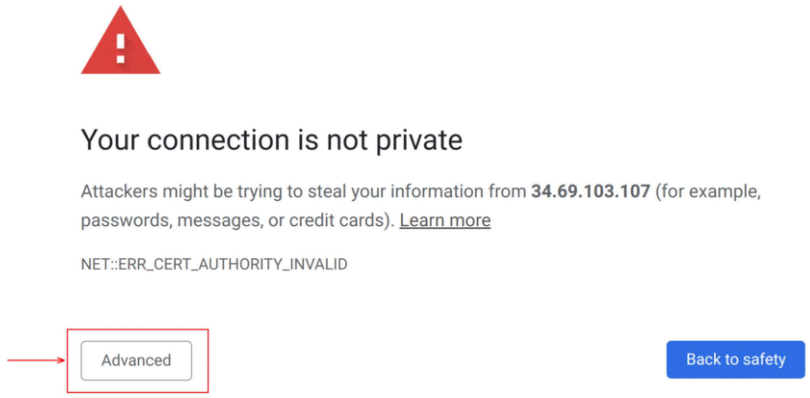
As usual, this command does not print any output if it succeeds. We can verify that it worked by actually visiting the website.

Accessing the web interface

Step 5: You can now access your website. To do that copy-paste the external IP address of **backup-server** in a new browser window. Use the format

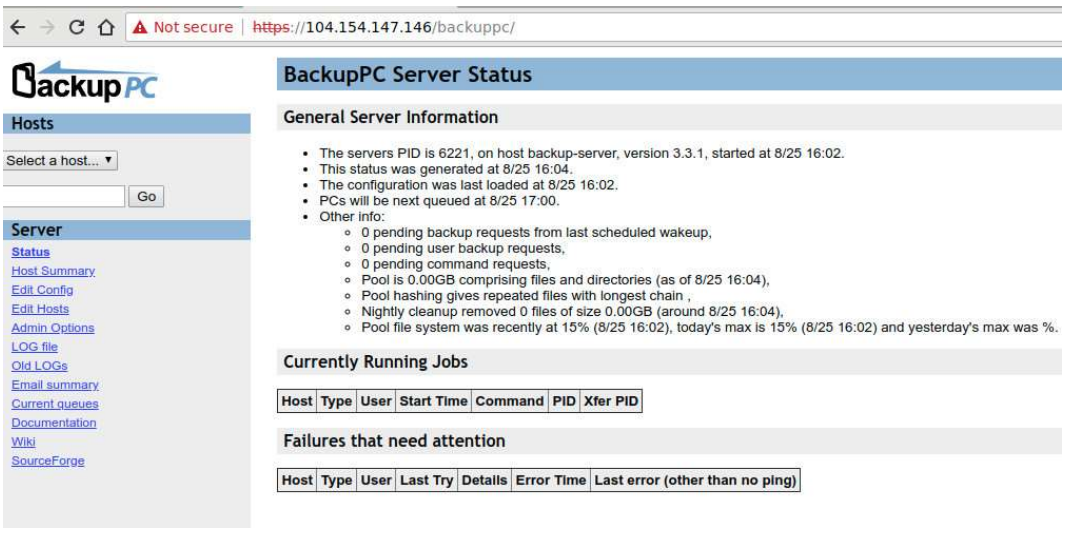
https://[backup-server external IP address], and replace `[backup-server external IP address]` with the external IP address of **backup-server**, and press "Enter". You can find the external IP address for `backup-server` in Connection Details Panel.

The browser will warn you that the SSL certificate being used is not trusted. This is expected because we have not registered our machine with a Certificate Authority. If this were a machine that we owned instead of a lab machine, we would request the certificate properly so that the browser would detect it as valid.



For this lab, please accept the SSL certificate even if your browser warns you that it's invalid or self-signed. Once you do that, you'll access the default SSL website. In order to access the BackupPc administrative interface you need to append "`/backuppc`" to the URL (e.g. `https://10.20.30.40/backuppc/`)

When you do that, your browser will ask you for your username and password. The username is `backuppc` and the password is the one you entered with the `htpasswd` command.

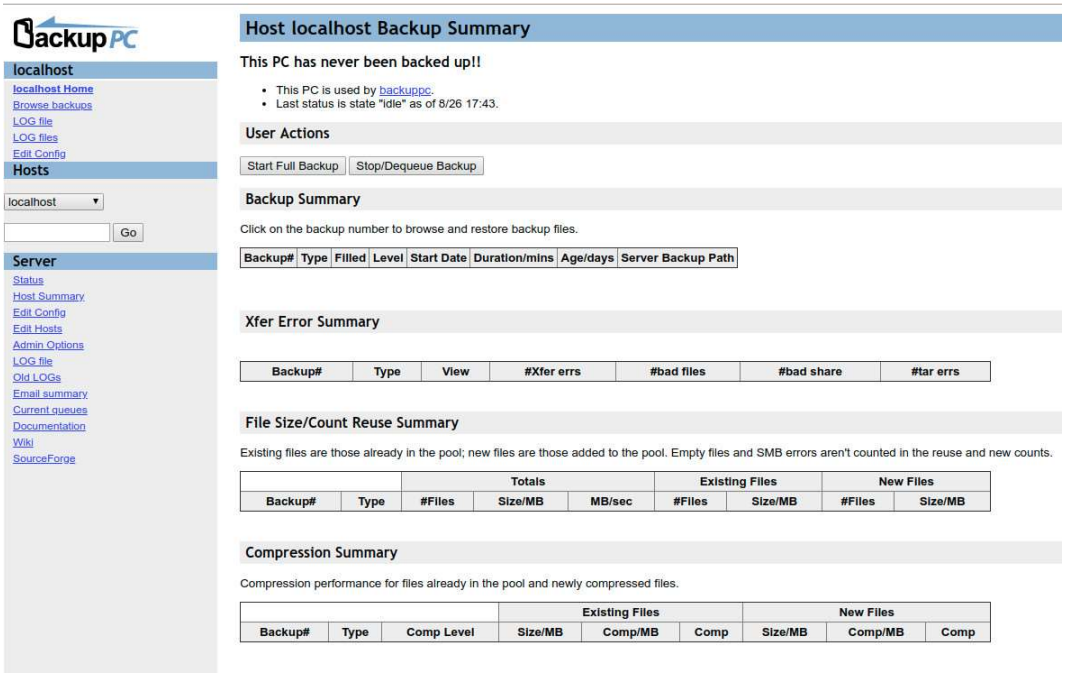


We can now administer our BackupPc installation through the web. Feel free to browse around the website and have a look at the configuration options that are available.

Backing up the /etc directory of localhost

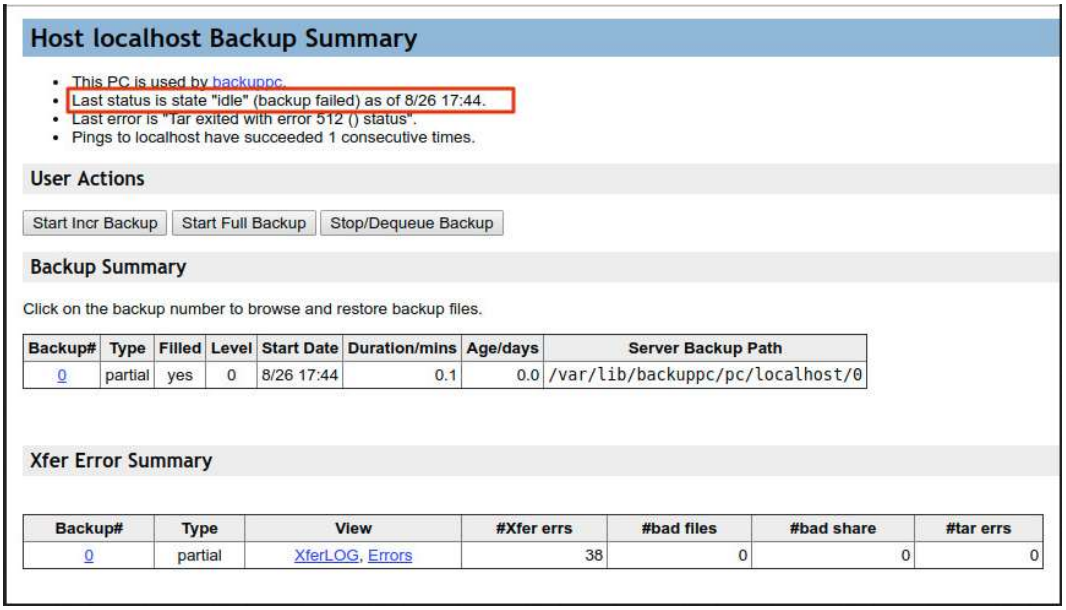
We will now start by creating a backup of the files in the local machine. To do that, first click on the **Host Summary** link. The host summary page lists all the currently known hosts and their state. For now, the only known host to the backup system is `localhost`. It will be listed under **Hosts with no Backups** because it doesn't yet have any backups.

If you click on the **localhost** link, it will open the page with the backup information corresponding to the local machine.



This page tells us that the machine has nothing backed up yet. To change that, we can click on the **Start Full Backup** button. This will ask for confirmation that this is what we really want and after that, provide a link to go back to the homepage for the host.

Click on that link and you'll be back at the summary page, but this time it will say something different. If it says that the backup is in progress, wait a few seconds and reload the page until it says that the status is **idle**.



The message at the top of the page says that the backup failed and that there were errors. In the summary section we can see that a backup was attempted but it says that it's "partial". This means that the backup couldn't complete. Let's look at the errors, by clicking on the **Errors** link in the "Backup 0" line.

This page shows all the errors that occurred while attempting to do the backup. It's a long page of error messages.

File /var/lib/backuppc/pc/localhost/XferLOG.0.z (Extracting only Errors)

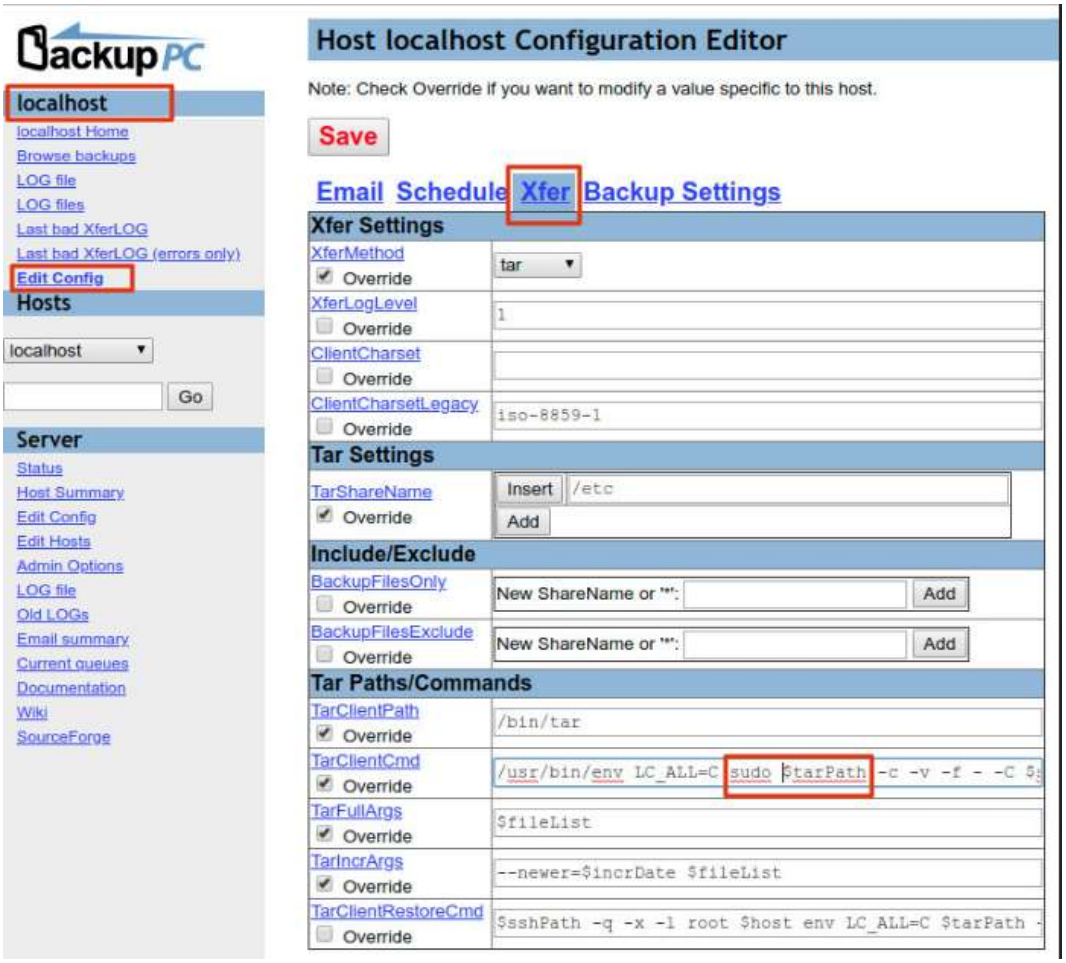
Contents of file /var/lib/backuppc/pc/localhost/XferLOG.0.z, modified 2018-08-26 17:44:10 (Extracting only Errors)

```
Running: /usr/bin/env LC_ALL=C /bin/tar -c -v -f - -C /etc --totals .
full backup started for directory /etc
Xfer PIDs are now 6410,6409
/bin/tar: ./at.deny: Cannot open: Permission denied
/bin/tar: ./ssl/private: Cannot open: Permission denied
[ skipped 519 lines ]
/bin/tar: ./subgid-: Cannot open: Permission denied
/bin/tar: ./sudoers: Cannot open: Permission denied
/bin/tar: ./group-: Cannot open: Permission denied
/bin/tar: ./ssh/ssh_host_rsa_key: Cannot open: Permission denied
/bin/tar: ./ssh/ssh_host_dsa_key: Cannot open: Permission denied
/bin/tar: ./ssh/ssh_host_ed25519_key: Cannot open: Permission denied
[ skipped 123 lines ]
/bin/tar: ./ssh/ssh_host_ecdsa_key: Cannot open: Permission denied
/bin/tar: ./gshadow: Cannot open: Permission denied
/bin/tar: ./gshadow-: Cannot open: Permission denied
[ skipped 148 lines ]
/bin/tar: ./shadow-: Cannot open: Permission denied
/bin/tar: ./iscsi/initiatorname.iscsi: Cannot open: Permission denied
/bin/tar: ./iscsi/iscsid.conf: Cannot open: Permission denied
[ skipped 284 lines ]
/bin/tar: ./ufw/user6.rules: Cannot open: Permission denied
/bin/tar: ./ufw/before.init: Cannot open: Permission denied
/bin/tar: ./ufw/after.rules: Cannot open: Permission denied
/bin/tar: ./ufw/after6.rules: Cannot open: Permission denied
/bin/tar: ./ufw/user.rules: Cannot open: Permission denied
/bin/tar: ./ufw/before.rules: Cannot open: Permission denied
/bin/tar: ./ufw/before6.rules: Cannot open: Permission denied
/bin/tar: ./ufw/after.init: Cannot open: Permission denied
[ skipped 113 lines ]
/bin/tar: ./apparmor.d/cache/usr.sbin.ntpd: Cannot open: Permission denied
/bin/tar: ./apparmor.d/cache/usr.lib.lxd.lxd-bridge-proxy: Cannot open: Permission denied
/bin/tar: ./apparmor.d/cache/usr.lib.snapd.snap-confine.real: Cannot open: Permission denied
/bin/tar: ./apparmor.d/cache/usr.bin.lxc-start: Cannot open: Permission denied
/bin/tar: ./apparmor.d/cache/sbin.dhclient: Cannot open: Permission denied
/bin/tar: ./apparmor.d/cache/lxc-containers: Cannot open: Permission denied
/bin/tar: ./apparmor.d/cache/usr.sbin.tcpdump: Cannot open: Permission denied
/bin/tar: ./polkit-1/localauthority: Cannot open: Permission denied
[ skipped 120 lines ]
/bin/tar: ./sudoers.d: Cannot open: Permission denied
[ skipped 132 lines ]
/bin/tar: ./subuid-: Cannot open: Permission denied
[ skipped 143 lines ]
/bin/tar: ./passwd-: Cannot open: Permission denied
/bin/tar: ./security/opasswd: Cannot open: Permission denied
/bin/tar: ./shadow: Cannot open: Permission denied
[ skipped 339 lines ]
/bin/tar: ./.pwd.lock: Cannot open: Permission denied
[ skipped 1 lines ]
/bin/tar: Exiting with failure status due to previous errors
Tar exited with error 512 () status
[ skipped 143 lines ]
tarExtract: Done: 0 errors, 3 filesExist, 0 sizeExist, 0 sizeExistComp, 1856 filesTotal, 2246850 sizeTotal
Got fatal error during xfer (Tar exited with error 512 () status)
Backup aborted (Tar exited with error 512 () status)
Saving this as a partial backup, replacing the prior one (got 1856 and 2064 files versus 0)
```

There are many errors, but the message is always the same: **Permission denied**. This error is caused by the fact that the backup is running with the `backuppc` user, and that user doesn't have permission to see many of the files and directories inside the `/etc` directory.

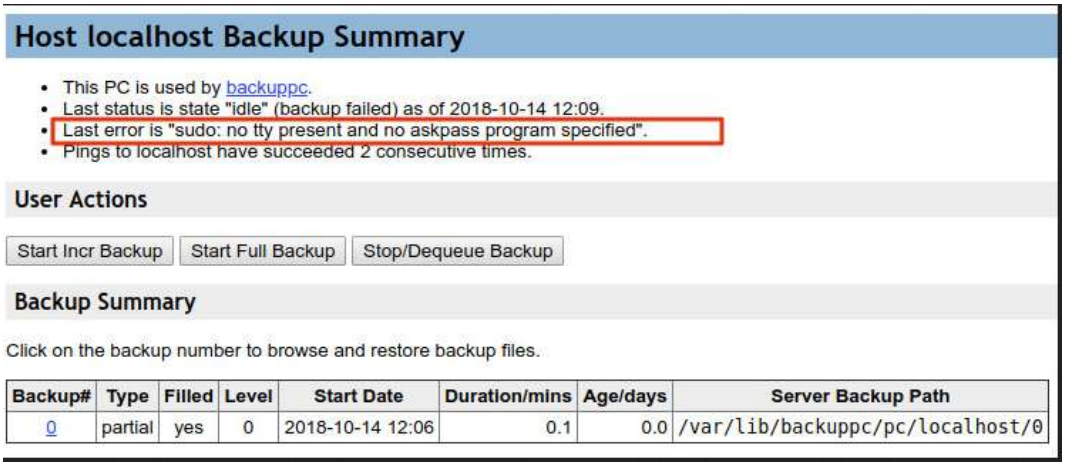
Let's have a look at how these files are being backed up. Click on the **Edit Config** link under the **localhost** section on the left hand side. Then click on the **Xfer** (short for transfer) tab.

This is the configuration related to how the backup should be obtained and what resources should be backed up. We can see for example that the **XferMethod** is `tar`, which is a very common linux command used for creating bundles of files. We can also see in the **TarShareName** option that the directory being backed up is `/etc`.



In order to fix the permission problem that we encountered, we will tell `backuppc` to use the `sudo` command for the creation of the backup. To do this, in the **TarClientCmd** option, add `sudo` before `$tarPath`. This will allow the `tar` command to have access to all the files that `root` has access to.

Once you've added this, click on the **Save** button. Go back to the **localhost Home** page and click the **Start Full Backup** button again. Confirm and then go back to the home page. Again, reload until it says "idle".

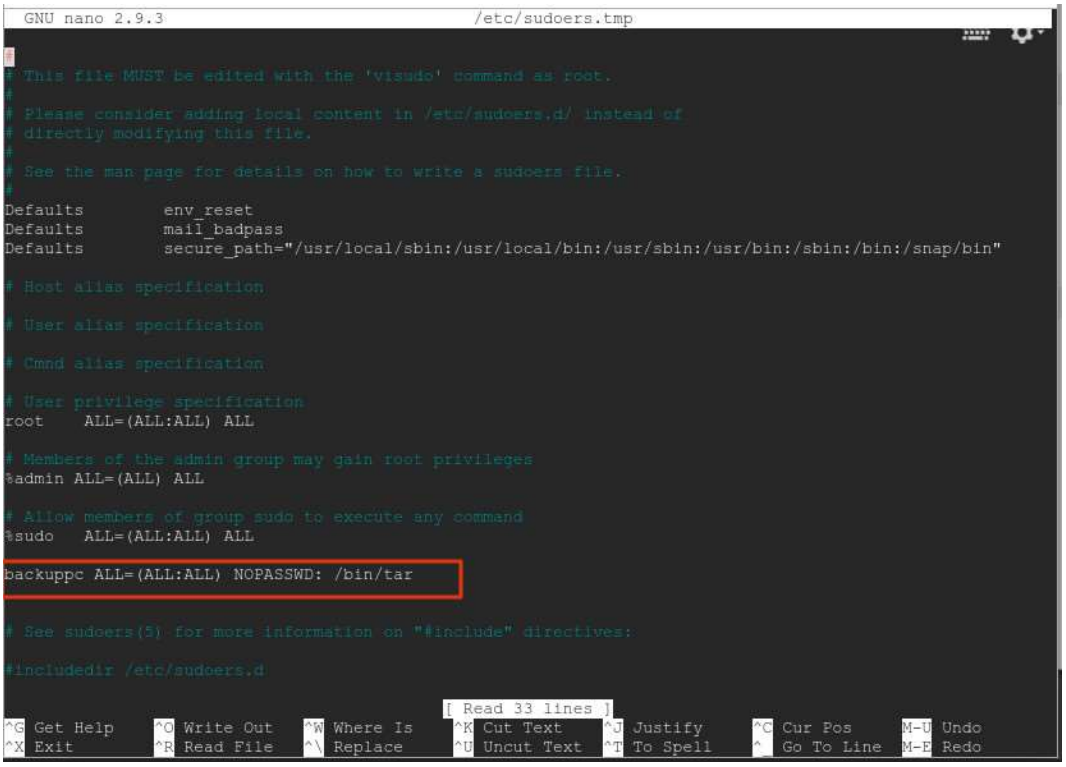


This time it should have failed with a different error: **sudo: no tty present and no askpass program specified**. This is caused by the fact that we requested to use the `sudo` command, but the `backuppc` user is not yet allowed to use it.

We've been using `sudo` all along these labs to execute commands with administrative priviledges. This works because our user is already allowed to execute any commands with `sudo`. We will now allow the `backuppc` user to execute `/bin/tar` (the command for creating the backup) using `sudo`, without requiring a password.

To do this, we'll edit the `/etc/sudoers` file. This file is special and can't be edited using the sudo command directly, we need to use a special command called `visudo`. Go to the **SSH window** of **backup-server** and execute the below command.

```
sudo visudo
```



To allow the `backuppc` user to execute `/bin/tar` unconditionally and without requiring a password, we need to add the following line to the file, after the other specifications:

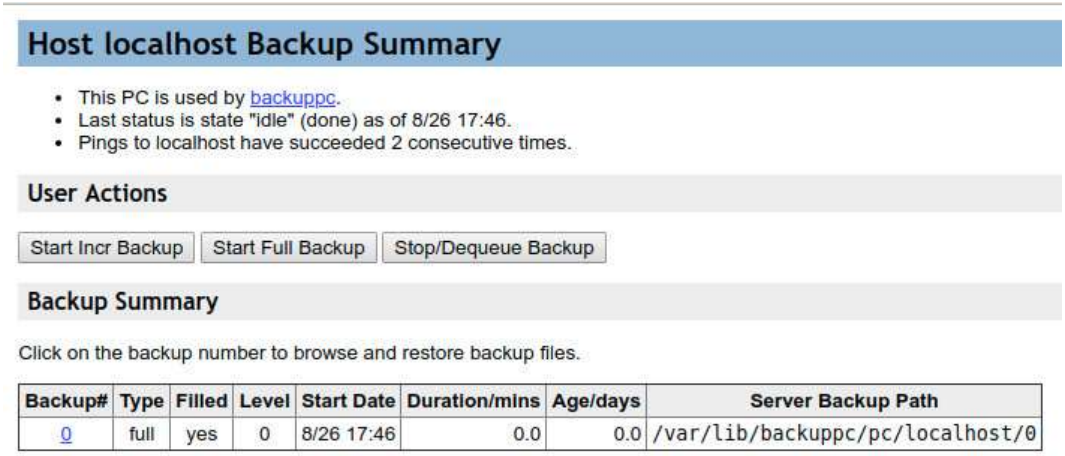
```
backuppc ALL=(ALL:ALL) NOPASSWD:/bin/tar
```

This line means that the `backuppc` user is allowed to use `sudo` to become any user of any group, it will not be prompted for a password, but it's only allowed to run `/bin/tar` as the command. We allow only this command both to avoid mistakes that could be problematic and to avoid granting unnecessary permissions.

Once you've edited the file, press **Ctrl-X** to exit, answer **Y** at the prompt to save changes and then press **ENTER** at the file name prompt.

With that, the `backuppc` user should now be allowed to access all files when creating the backup. Go back to the **localhost Home** page and click the **Start Full**

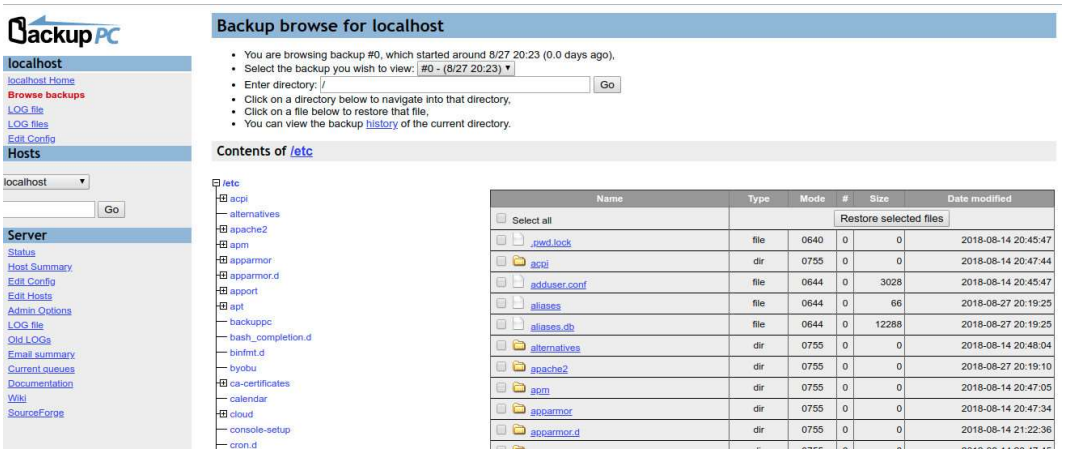
Backup button again. Confirm and then go back to the home page.



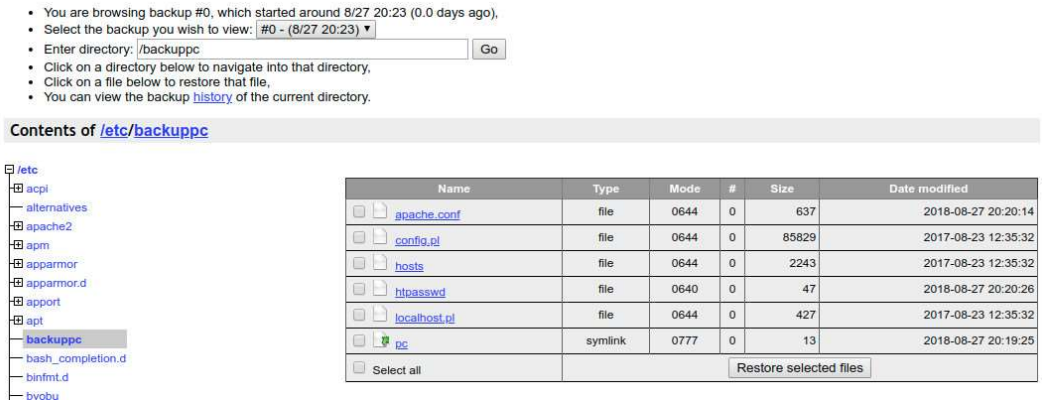
This time the lines at the top tell us that the backup is done. We can see that the **Type** of the backup is now **full**, which means that it completed successfully.

Yay! First backup done and with a tricky permission problem fixed along the way!

You can see the contents of any generated backup by clicking on the link of the backup entry (**0** in this case). You can also go directly to the latest backup by clicking **Browse backups** in the top left menu.



As you can see, the backups created by BackupPc can be browsed directly. You can click on the links to navigate the tree that was backed up. For example, if you click on the **backuppc** link, you'll see the files contained in that directory.



To inspect the contents of a backup, you can directly download any files on the list by clicking on their names. For example, you can click on **apache.conf** and you will be prompted to download the file in your machine. You can then open it and verify that it contains the same contents that we saw before.

If you need to restore something from a backup, you can select the items that you want to restore by clicking the checkbox on the left of their names and then clicking the **Restore selected files** button. This will let you select how you want to actually restore those files: you can either directly restore the files onto the drive or you can download a Zip or a Tar file containing the files.

Congratulations! By now, you've configured a backup server, performed a full backup fixing an error along the way and verified that the backups work correctly by downloading one of the backed up files. This is already a lot of progress but there are more interesting things to come.

Click *Check my progress* to verify the objective.

Backup localhost /etc directory

Backing up a remote Linux machine

As we mentioned earlier, we want to setup our `backup-server` so that it can backup not only the contents of the local machine, but also data stored in other machines, both Linux and Windows machines.

To allow our server to connect to the Linux instance that we want to back up, we will create an SSH key pair that will be stored in the machines.

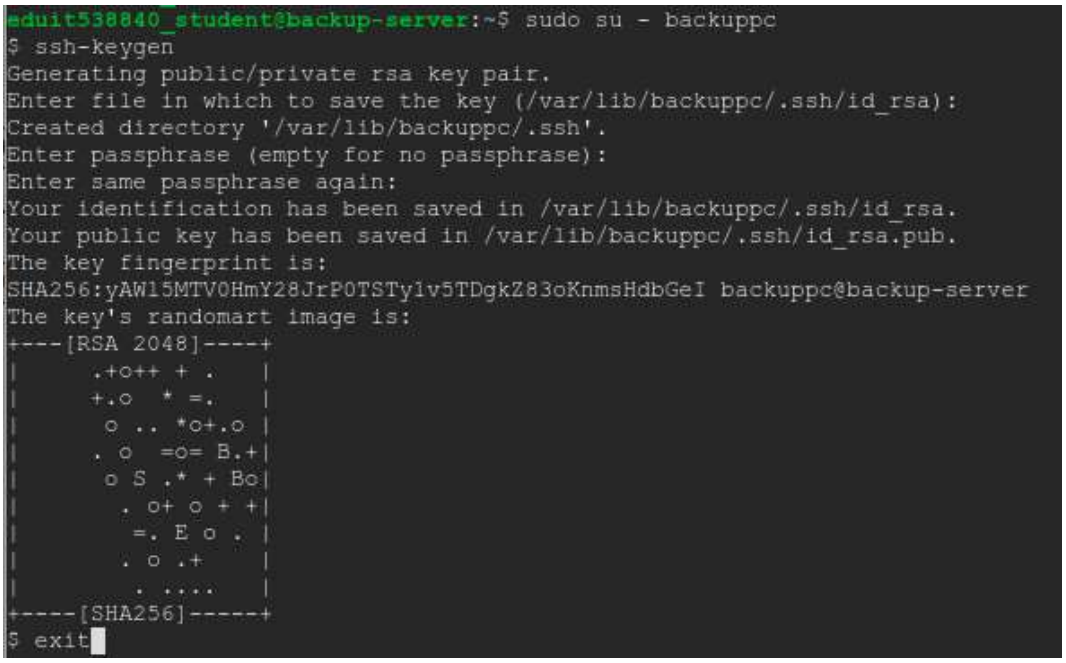
Note: You may have noticed that when you connect to Linux machines inside Google Cloud there's a message stating that the SSH keys are being created and then transferred. We are now going to replicate this process, for our "backuppc" user.
First, let's switch into the `backuppc` user:

```
sudo su - backuppc
```

The `su` command (stands for switch user) allows us to become a different user. In this case it's the `backuppc` user, who will be used to perform the backups. The next command will be executed by that user instead of ours.

Now, we will create an SSH key for the `backuppc` user:

```
ssh-keygen
```



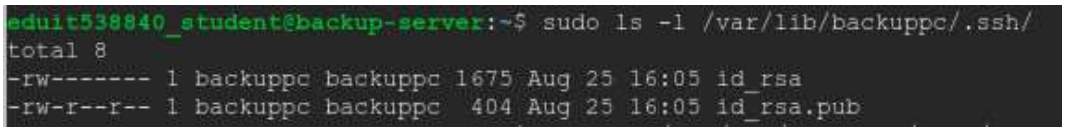
The `ssh-keygen` command generates an SSH key pair, that can be used to authenticate a user without having to provide an interactive password. The command will request the location of the file to be created and then will request a password. Simply press **ENTER** on the three prompts you get to use the default path and an empty password.

Once that's done, you can go back to your user by using the `exit` command.

```
exit
```

Let's look at the contents of the directory where the keys were generated.

```
sudo ls -l /var/lib/backuppc/.ssh/
```



The SSH key pair is composed by `id_rsa` and `id_rsa.pub`, the first one is the private key and the second one is the public key. This method of authenticating machines is called **public key cryptography** and will be explained in detail in the IT Security course. For now, you just need to know that we want to copy the contents of the public key (`id_rsa.pub`) to any machines to which we want the `backuppc` user to be able to authenticate.

To do this, copy the file to `/tmp`, so that it can be accessed without administrator rights:

```
sudo cp /var/lib/backuppc/.ssh/id_rsa.pub /tmp/
```

And then copy it to `linux-instance` using the following command:

```
gcloud compute scp /tmp/id_rsa.pub root@linux-instance:~ --zone=us-central1-a
```

Press **ENTER** for any prompts.

Once you have copied the file, connect to `linux-instance` using its external IP address and the local SSH client. `linux-instance external IP address` can be found at Connection Details Panel. Follow the instructions given in the section `Accessing the linux virtual machine`. Click on `Accessing the linux virtual machine` from the navigation pane at the right side.

In that SSH session, find the file you copied using the following command:

```
sudo find / -type f -name "id_rsa.pub"
```

```
gcpstagingeduit1625_student@linux-instance:~$ sudo find / -type f -name "id_rsa.pub"
/home/sa_101916025614613369929/id_rsa.pub
gcpstagingeduit1625_student@linux-instance:~$
```

Note down the path of the file as shown in the above image.

Now move this file to your home directory using the following command. Replace `[path_to_key_file]` with the path noted down earlier. Also, replace the `[username]` with the one provided in Connection Details Panel on the left side.

```
sudo mv [path_to_key_file] /home/[username]
```

The file is now in your home directory. In order for the connection to work automatically, we need the contents of `id_rsa.pub` to be added to the `authorized_keys` file for the `root` user.

The `authorized_keys` file is used to indicate which SSH keys are allowed to login as a user of a machine without entering additional credentials. In this case, we want the user that owns this key (the `backuppc` user on the `backup-server` instance) to be able to login as `root` on the `linux-instance` machine.

To put the contents in the right place, we will use a command called `tee` that redirects the input that it receives to both the console and a file.

```
cat id_rsa.pub | sudo tee -a /root/.ssh/authorized_keys
```

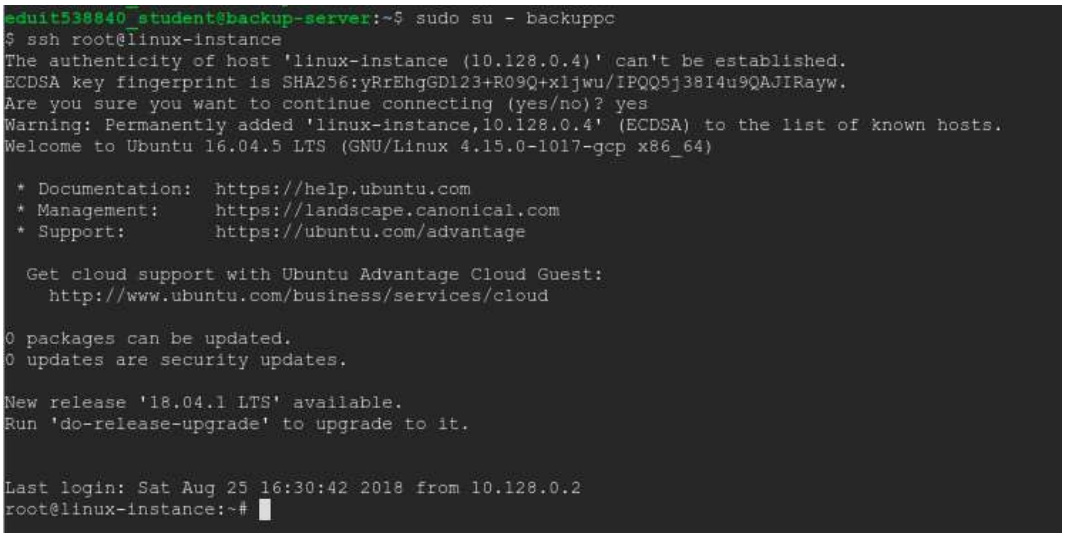
```
eduit538840_student@linux-instance:~$ cat id_rsa.pub | sudo tee -a /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDP0fcb9i8RESbHuftozSa09Z20v49/IuDimR6jYSrITCvC3VW1EmfSQoQ6691ZB5LOPsvWcM3k82e
Q6wBa7V2PJz1xNJPVtWhSc4PqD/E/7HrwdpkQaI2qDfdzEza/W2x6ZxoPUTaqbhZgIeOpOq6qndXJSWUjX+b/wAjNHrC6xBHnmHVi1OOLbkLhPLFWsg
4ND9NvZOoX/i1bIB++VczRF+YFDABIQ15kWVz4wn4oSTzxQcQrDY34UWOhjVaa8xULvW/zHSz7nzRAUJY3mKpw1OZzXu2DeJDS3w3jWTo2oFQeKfnDk
Ewzqf8jLTyIzNM8M3JggHrobwK696yN0hiD backuppc@backup-server
```

The command above is printing the contents of `id_rsa.pub` to your screen while also appending it to `/root/.ssh/authorized_keys`, which is where we want it to be.

With that set, our `backuppc` user should be able to login as `root` in `linux-instance`. Let's verify that this works correctly.

Go back to your **backup-server SSH session** (you can distinguish between them by looking at the hostname in the green prompt). You need to become the `backuppc` user by using the `su` command. Then use the `ssh` command to connect to `linux-instance`.

```
sudo su - backuppc
ssh root@linux-instance
```



When you first connect to a machine using `ssh`, you are prompted to verify the **"fingerprint"** of that machine's certificate. The fingerprint is a short version of a machine certificate that can be used to ensure that you are connecting to the right machine. There are more details about this in the IT Security course. For now, what you need to know is that this identifies the machine in a secure way. Any future connection to that machine will check that the fingerprint matches the stored one. You will receive a warning if that's not the case.

Once you've answered "`yes`" to the fingerprint prompt, the machine should let you connect as root. If this doesn't work, then something went wrong during the setup for the SSH key exchange. Review the steps above and verify that you didn't skip any.

After you've verified that the SSH connection is working properly, close the SSH session by typing `logout`.

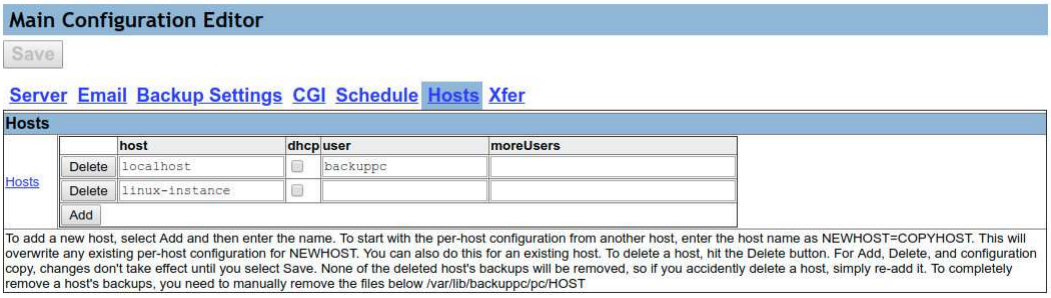
```
logout
```

Then close the `backuppc` session by typing `exit`.

```
exit
```

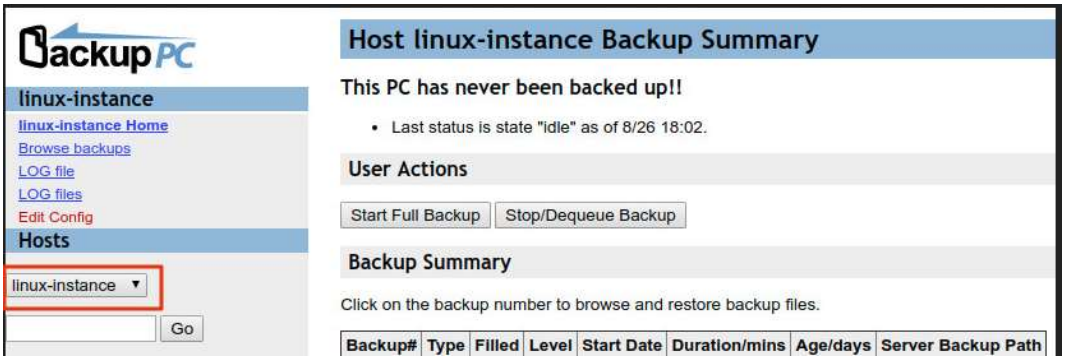
Adding the remote machine in BackupPc

Now that we have the remote connection between the machines working, you can configure this new client in BackupPc. Go back to the administrative web interface and click on **Edit Hosts**.

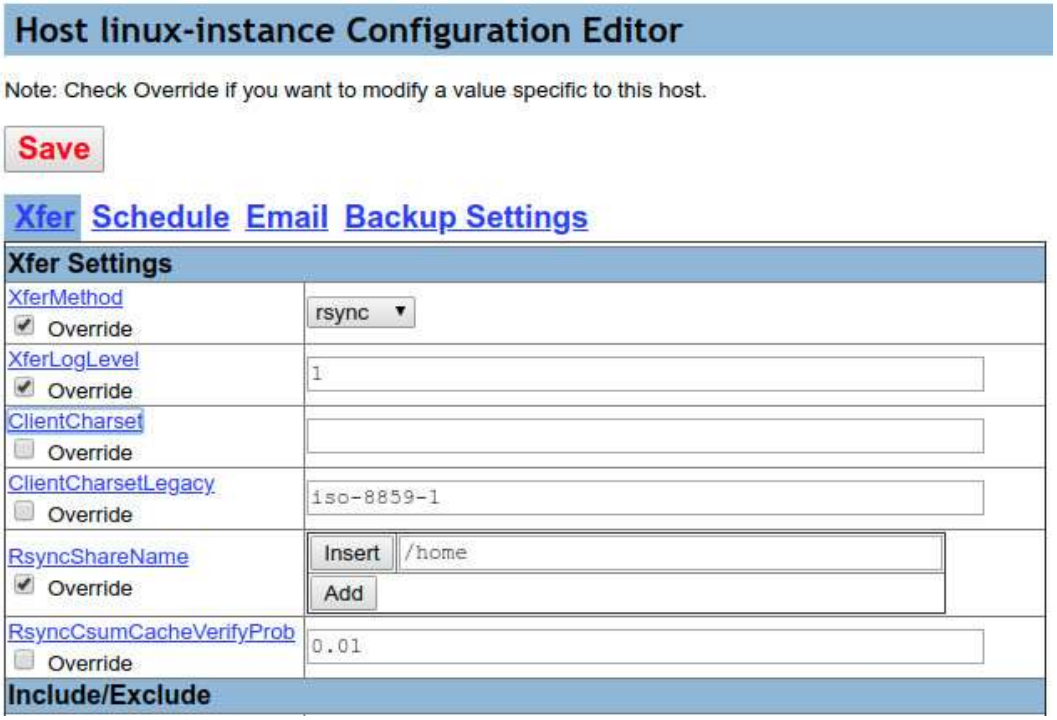


We will add `linux-instance` as a new host. To do that, click the **Add** button to add a new line, and enter `linux-instance` in the **host** field. Then click the **Save** button.

We've added the host, but we haven't configured it yet. Let's do that by clicking **Host summary**, which now shows us the two hosts known to the system (one with backups and the other one without) and then clicking **linux-instance** to go to the settings page of this instance.



The top part of the left bar now refers to this host that we want to configure. In order to do that, click **Edit Config** and then click the **Xfer** tab.



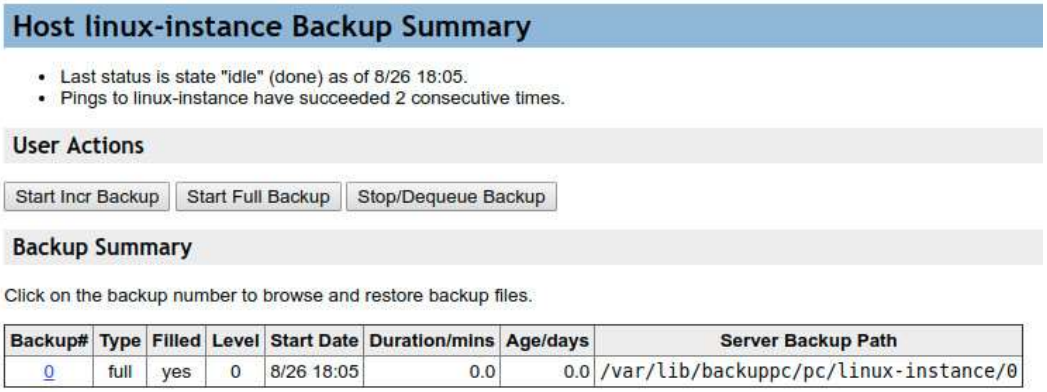
There are many different transfer methods that can be used to perform backups. Each of them has advantages and disadvantages. You can read more about the different methods in BackupPc's documentation.

For this instance, please select **rsync** as the **XferMethod**. `rsync` is a tool that can be used to synchronize the contents of two directory trees without transferring the whole contents. The program verifies what has already been transferred and only transfers new data.

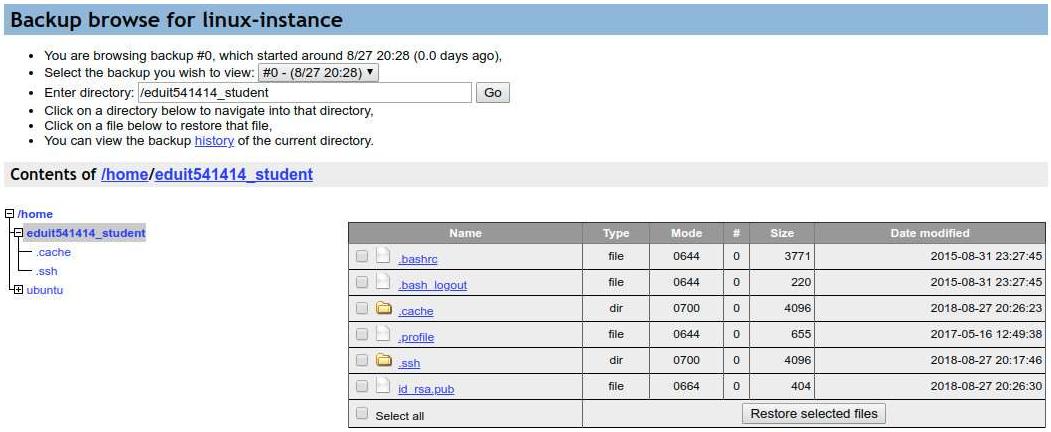
In the path to backup (called **RsyncShareName** in the `rsync` configuration) enter `/home`. This is the remote directory that we want to backup. We could also backup the whole machine by entering `/`, but in our scenario we care only about `/home`, which contains the home directories for the users.

Once you've entered this, click the **Save** button to save your changes. Then go to the **linux-instance Home** link, which shows you the current state of the backups for this machine. In this case, there aren't any backups yet.

Click the **Start Full Backup** button to force a new backup for this machine to start. Once you've accepted the confirmation prompt, go back to the home page.



If everything went right, you should see that the backup was completed. As with the previous case, you can now browse this backup to see that it actually worked.



If you look into the folder for your user, you'll see that the `id_rsa.pub` file that we copied a few minutes ago is part of the list of backed up files.

Let's try deleting it and then restoring it. Go back to your **linux-instance SSH session** and delete the `id_rsa.pub` file using the `rm` command:

```
rm id_rsa.pub
```

In case of any prompts, type **yes**.

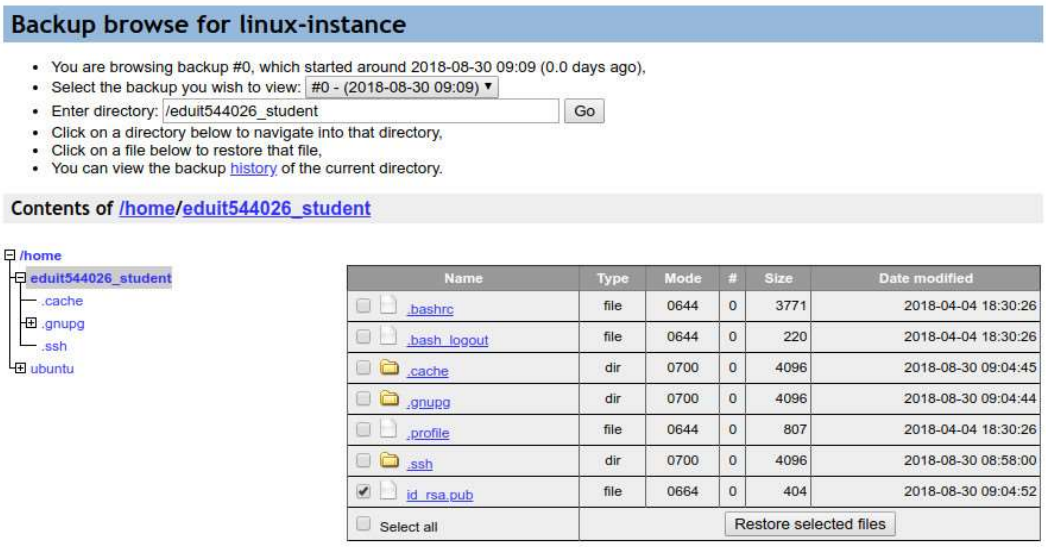
This command doesn't print anything, but we can see that it worked by listing the contents of the current directory.

```
ls -la
```

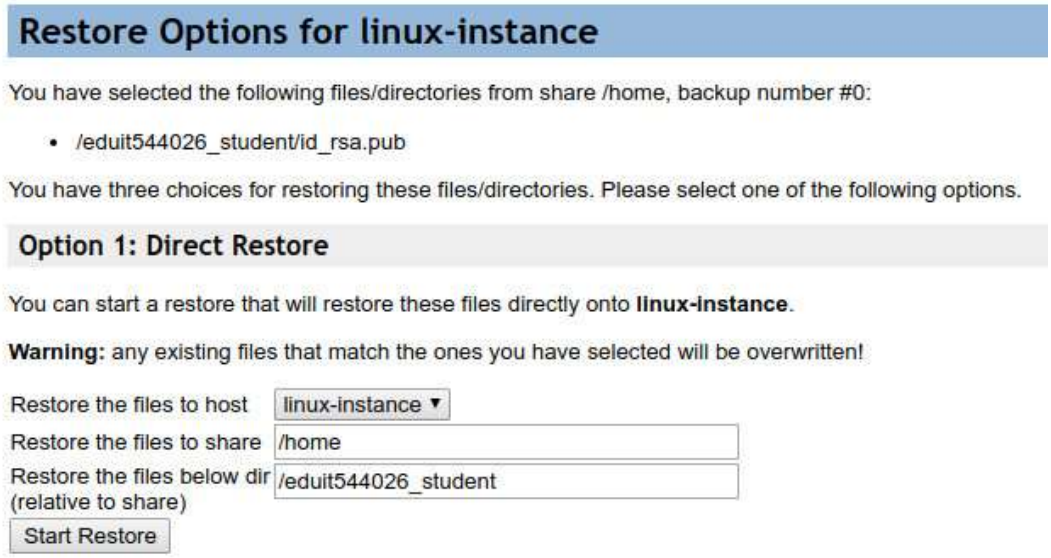
```
eduit544026_student@linux-instance:~$ rm id_rsa.pub
eduit544026_student@linux-instance:~$ ls -la
total 32
drwxr-xr-x 5 eduit544026_student eduit544026_student 4096 Aug 30 09:10 .
drwxr-xr-x 4 root                root                4096 Aug 30 08:58 ..
-rw-r--r-- 1 eduit544026_student eduit544026_student  220 Apr  4 18:30 .bash_logout
-rw-r--r-- 1 eduit544026_student eduit544026_student 3771 Apr  4 18:30 .bashrc
drwx------ 2 eduit544026_student eduit544026_student 4096 Aug 30 09:04 .cache
drwx------ 3 eduit544026_student eduit544026_student 4096 Aug 30 09:04 .gnupg
-rw-r--r-- 1 eduit544026_student eduit544026_student  807 Apr  4 18:30 .profile
drwx------ 2 eduit544026_student eduit544026_student 4096 Aug 30 08:58 .ssh
```

Reminder: The "-a" flag allows us to also see hidden files (the ones with names that start with a dot). We can see all the files that are in the directory. This

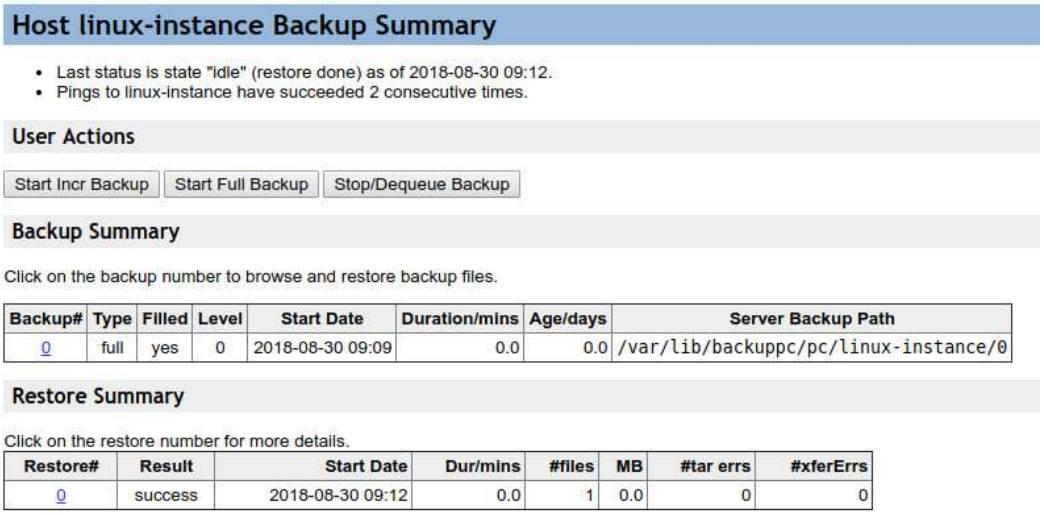
matches the list that we saw when browsing the backup, and there is no "id_rsa.pub" because we deleted that one.
Now that the file has been deleted, let's recover it by checking the box next to its name and then pressing the **Restore selected files** button on the BackupPc interface.



When you press this button, you are taken to the **Restore Options** page. Here you can select if you want to download the file or do a direct restore. In this case, we want to do a direct restore.



Click the **Start Restore** button in the Direct Restore section. Confirm that this is what you want and go back to the home page.



Host linux-instance Backup Summary

- Last status is state "idle" (restore done) as of 2018-08-30 09:12.
- Pings to linux-instance have succeeded 2 consecutive times.

User Actions

Start Incr Backup | Start Full Backup | Stop/Dequeue Backup

Backup Summary

Click on the backup number to browse and restore backup files.

Backup#	Type	Filled	Level	Start Date	Duration/mins	Age/days	Server Backup Path
0	full	yes	0	2018-08-30 09:09	0.0	0.0	/var/lib/backuppc/pc/linux-instance/0

Restore Summary

Click on the restore number for more details.

Restore#	Result	Start Date	Dur/mins	#files	MB	#tar errs	#xferErrs
0	success	2018-08-30 09:12	0.0	1	0.0	0	0

The interface claims that it succeeded in restoring the file. Let's check if that's true by listing the contents of the directory and then showing the contents of the file.

`ls -la`

```
eduit544026_student@linux-instance:~$ ls -la
total 36
drwxr-xr-x 5 eduit544026_student eduit544026_student 4096 Aug 30 09:12 .
drwxr-xr-x 4 root                root                4096 Aug 30 08:58 ..
-rw-r--r-- 1 eduit544026_student eduit544026_student  220 Apr  4 18:30 .bash_logout
-rw-r--r-- 1 eduit544026_student eduit544026_student 3771 Apr  4 18:30 .bashrc
drwx------ 2 eduit544026_student eduit544026_student 4096 Aug 30 09:04 .cache
drwx------ 3 eduit544026_student eduit544026_student 4096 Aug 30 09:04 .gnupg
-rw-r--r-- 1 eduit544026_student eduit544026_student  807 Apr  4 18:30 .profile
drwx------ 2 eduit544026_student eduit544026_student 4096 Aug 30 08:58 .ssh
-rw-rw-r-- 1 eduit544026_student eduit544026_student  404 Aug 30 09:04 id_rsa.pub
```

`cat id_rsa.pub`

```
eduit544026_student@linux-instance:~$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDIrFfgL7M31HeUpSVPtMS22729qNwmkCJS0AkmiLL05PmRqEVE/ZaRz0On6hs4x11FMRTg46LuENK
Wg/zaS4VTBsBcObVKUWDFnXwsumNT6UeRONdotvL1FkbLkouDGGXloRN1kezSBVnpN8qubTpT1wY80ZYHDYp17RWz2n1CA5riXyX3//IYY1UTZhaA1/
4mJEnH/MR6+NDqU1y8/XzCAK21BV211q70H1bpAEaXIAO5Yb3aC+HZScwjLk6MvrG87wrHi8hrCtDz29/NIFoa5gCjG031h4badDtouhvGvOvF4fSNQ
2Xn2GHiV0mNTtSR/VkSZychHRyrqKmrTi6d backuppc@backup-server
```

Yay! Restores work as well!

Click *Check my progress* to verify the objective.

Backup linux-instance /home directory

Backing up a remote Windows machine

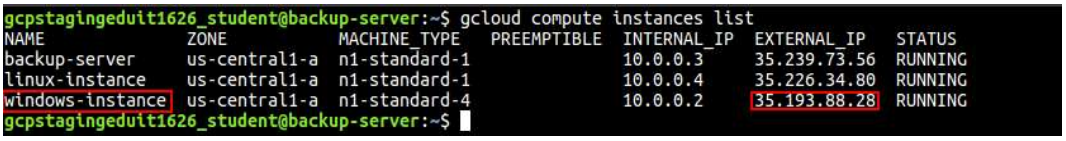
Finally, we want to also use our backup server to backup remote data from a Windows machine. In this case, the resource that we want to back up are the contents of the Users folder.

To make this possible, we will need to create a `backuppc` user that should be able to connect to the Users folder as a shared resource. First, we will connect to `windows-instance`.

To list all instances, use the following command, in either of the SSH sessions.

```
gcloud compute instances list
```

Note down the external IP address of `windows-instance`, as shown in the below image. It will be used during the lab.



Now connect to `windows-instance` using its external IP address and the local RDP client. Use the external IP address of `windows-instance` noted down earlier and follow the instructions given in the next section `Accessing the windows virtual machine`.

Accessing the windows virtual machine

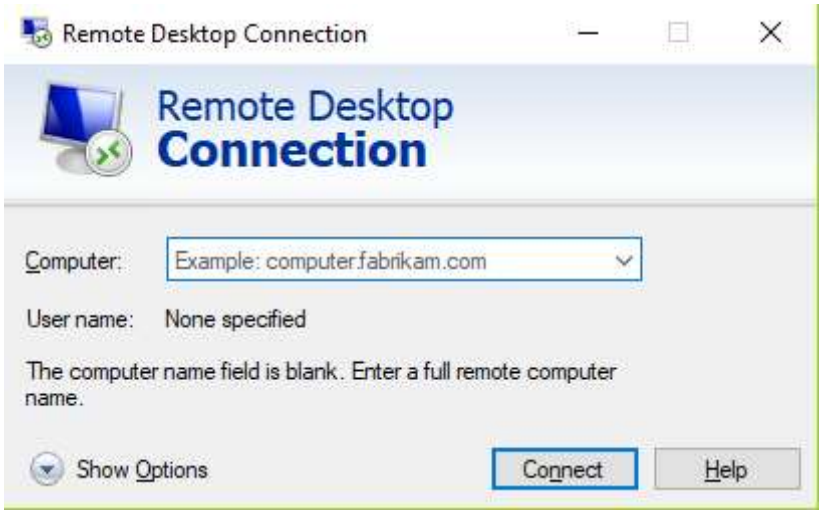
Please find one of the four relevant options below based on your device's operating system.

Option 1: Windows Users: Connecting to your VM via RDP

In this section, you will use Remote Desktop Connection to connect to your windows instance using its external IP address.

1. Open Remote Desktop Connection by clicking the **Start** button. In the search box, type Remote Desktop Connection, and then, in the list of results, click Remote Desktop Connection.

2. Enter the external IP address of the instance you want to connect to in the **Computer** field. Use the external IP address of `windows-instance` noted down during the lab. Click on **connect**.



3. Change the username to **student**. And use the password mentioned in the Connection Details Panel on the left side. Click **OK**.

4. Click **Yes** to accept the certificate.

You should now see a visual interface that looks exactly like the Windows 10 OS!

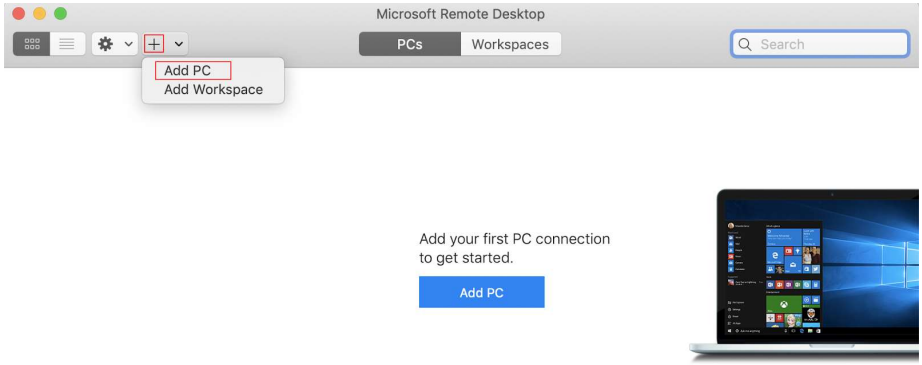
If you see any error message, close the window and wait a minute or so. Sometimes the VM-creation process takes a few minutes, and you won't be able to access the VM until it's finished. This also applies to any errors that say your credentials (username and password) are incorrect.

Option 2: OS X users: Connecting to your VM via RDP

In this section, you will use Microsoft Remote Desktop 10 to connect to your windows instance using its external IP address. OSX users can download Microsoft Remote Desktop from the Mac App Store. If you are using Microsoft Remote Desktop 8, note that the interface will vary slightly than what's listed below.

1. Open Microsoft Remote Desktop 10 application.

2. Click on + sign above, followed by **Add PC**.

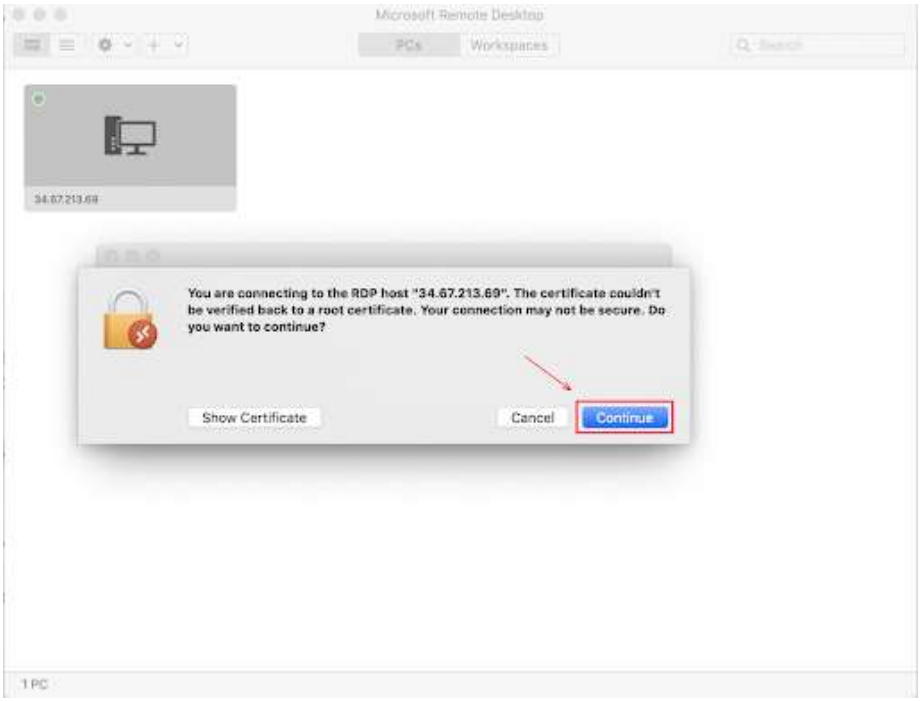


3. Enter the external IP address of the instance you want to connect to in the **PC name** field. Find the external IP address for your instance from the Connection Details Panel on the left side. Click on the **Add** button.

4. You should now be able to see your desktop represented by the external IP address of your VM instance under **PCs**. Double click on your VM's external IP address.

5. The application will now prompt you for username and password. Change the username to **student**. And use the password mentioned in the Connection Details Panel on the left side. Once you have entered the details click **Continue**.

6. For any prompt regarding 'Certificate verification', click **continue**.



You should now see a visual interface that looks exactly like the Windows 10 OS!

If you see any error message, close the window and wait a minute or so. Sometimes the VM-creation process takes a few minutes, and you won't be able to access the VM until it's finished. This also applies to any errors that say your credentials (username and password) are incorrect.

Option 3: Chrome OS users: Connecting to your VM via RDP

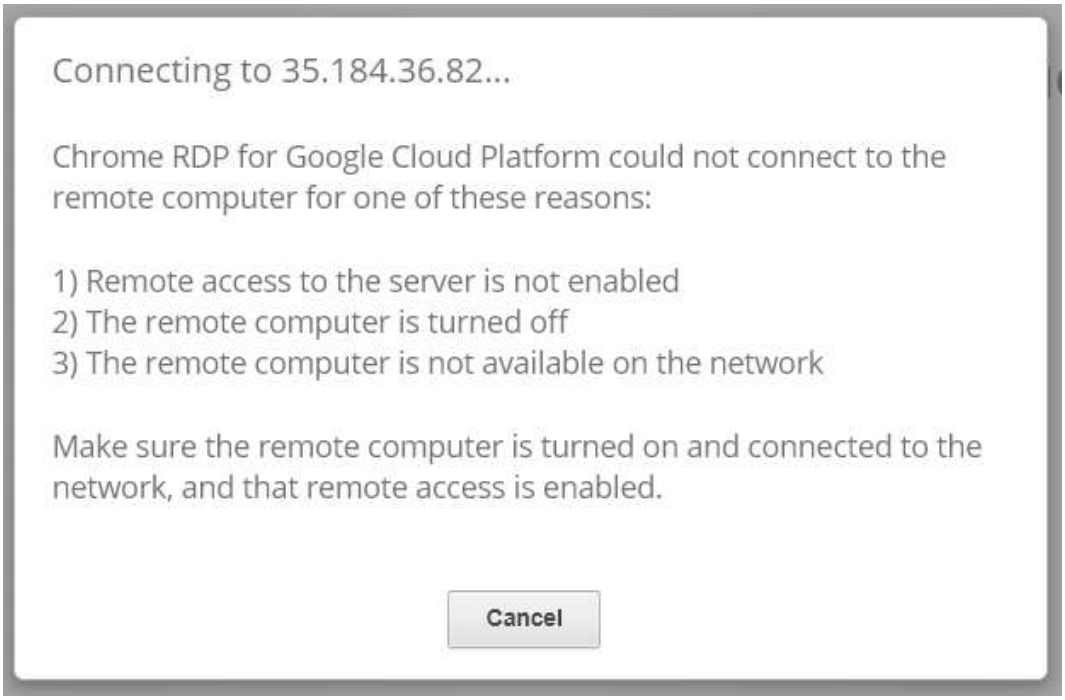
In this section, you will use Chrome RDP to connect to your windows instance using its external IP address.

Chrome OS users can download Chrome RDP from Chrome Web Store. Once you navigate to the download page, click on the **Add to Chrome** button. Click on **Add app** in case of any pop-ups. Then, click on **Launch app** to start the application.

1. Open the Chrome RDP application.
2. Enter the external IP address of the instance you want to connect to in the **Enter the computer name or address to connect to** field. Use the external IP address of `windows-instance` noted down during the lab. Click on **connect**.
3. Leave the domain field blank. Change the username to **student**. And use the password mentioned in the Connection Details Panel on the left side. Click **OK**.
4. Click **Continue** for any window related to certificate verification.

You should now see a visual interface that looks exactly like the Windows 10 OS!

If you see any error message (an example of one is shown below), close RDP and wait a minute or so. Sometimes the VM-creation process takes a few minutes, and you won't be able to access the VM until it's finished. This also applies to any errors that say your credentials (username and password) are incorrect.



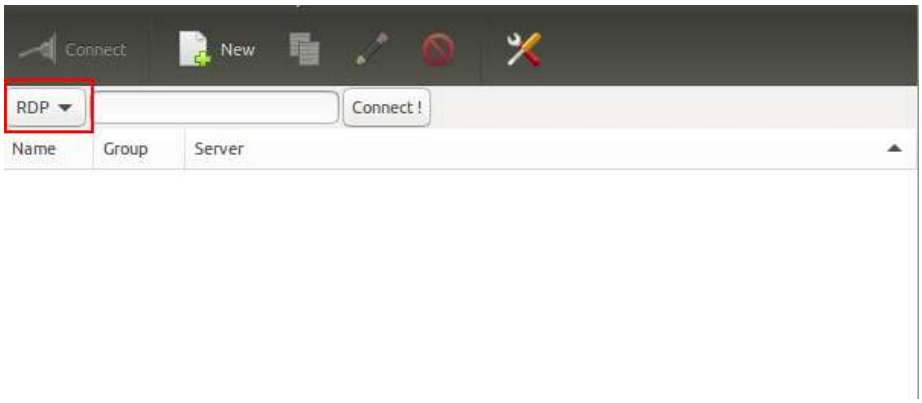
Option 4: Linux users: Connecting to your VM via RDP

In this section, you will use **Remmina** to connect to your windows instance using its external IP address. Open Remmina in your Linux machine. Linux users can install Remmina if it is not pre-installed.

1. Open Remmina.

2. Enter the external IP address of the instance you want to connect to. Use the external IP address of `windows-instance` noted down during the lab. Click on **Connect**.

   Make sure the connection protocol is set to **RDP**, as shown in the image below:



3. A window appears asking you accept the certificate, click **Ok** to continue.

4. Leave the domain field blank. Change the username to **student**. And use the password mentioned in the Connection Details Panel on the

left side.

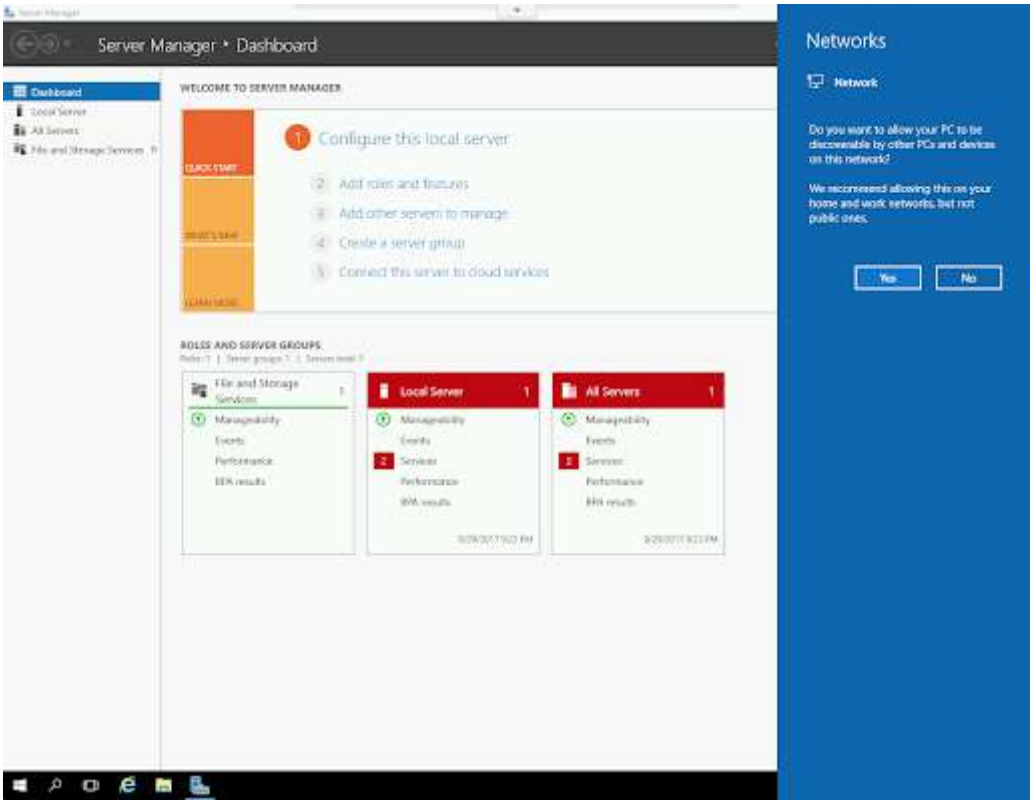
5. Enter the external IP address of the instance you want to connect to in the **Enter the computer name or address to connect to** field. Find the external IP address for your instance from the Connection Details Panel on the left side. Click on **connect**.

You should now see a visual interface that looks exactly like the Windows 10 OS!

If you see any error message, close the window and wait a minute or so. Sometimes the VM-creation process takes a few minutes, and you won't be able to access the VM until it's finished. This also applies to any errors that say your credentials (username and password) are incorrect.

Using the Windows instance

Now you have access to the Windows instance, you're ready to start using it! This version of Windows is intended to be used on a Server, and auto-starts a server-management program. We don't need this for this lab, so wait for it to finish starting and then close it. You may see the desktop appear for a few seconds before the program launches.

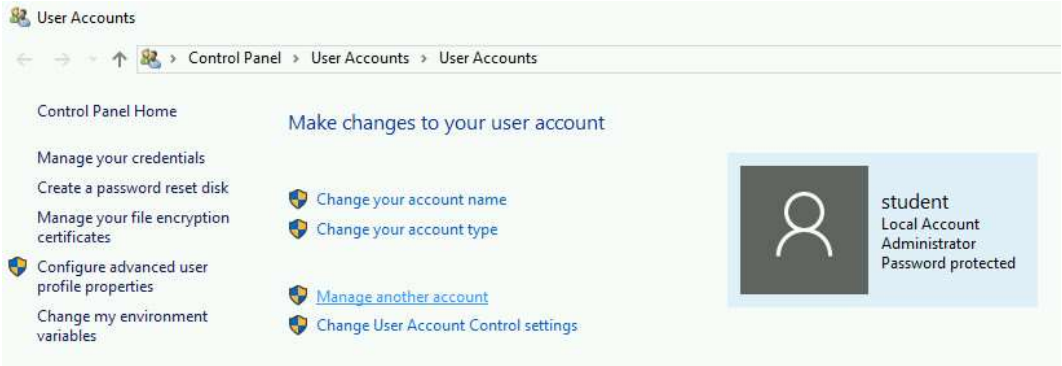


Once that's closed, the Windows OS is ready for you to use.

Finishing the login process

Now you‘ll see a Windows desktop background that looks like this:

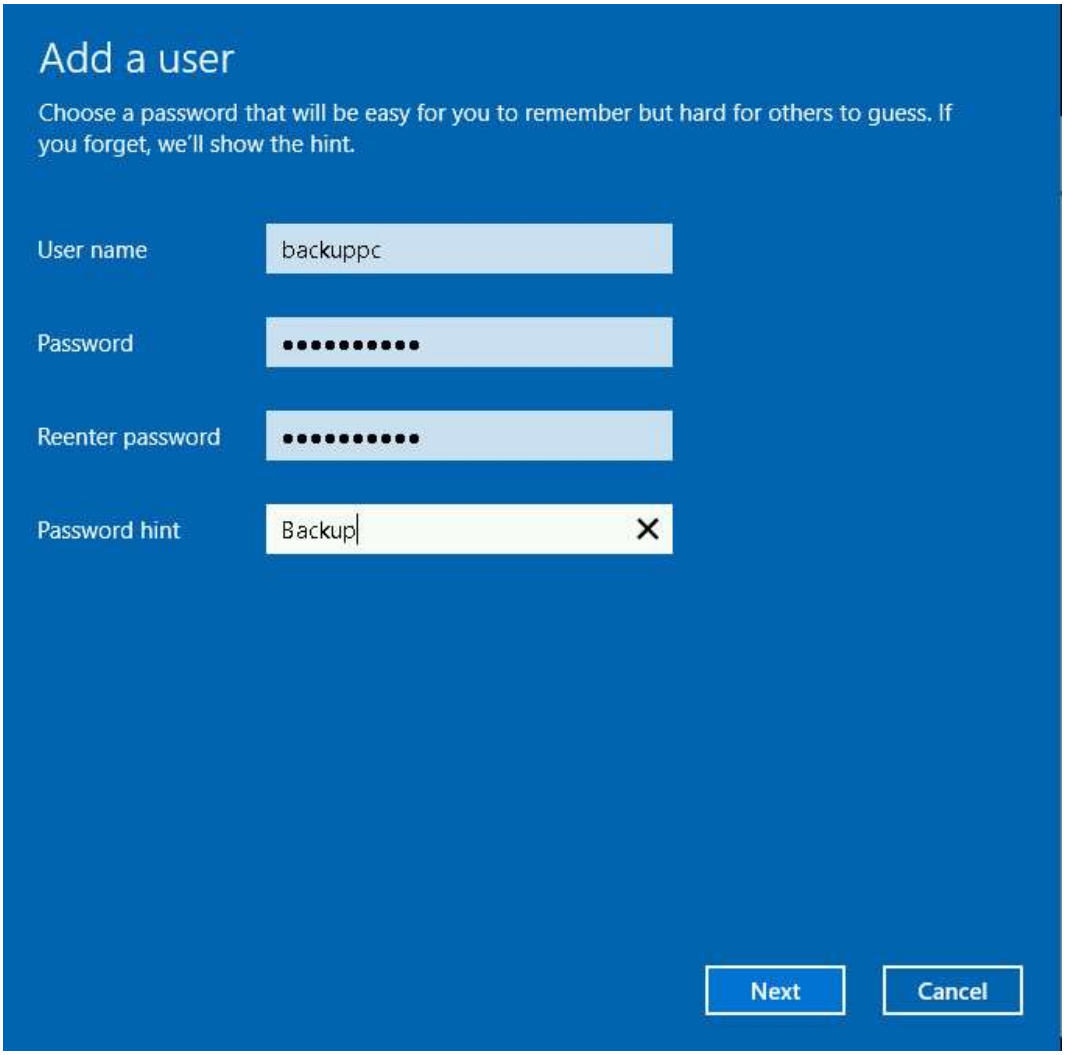
To create the new account, we want to open the **User Accounts** application. To do that, open the Control Panel, and then click on User Accounts and then again on User Accounts.



This is the application to manage local User Accounts on a Windows machine that is not managed with a centralized tool like Active Directory.

Click on **Manage another account** to be able to create a new account.

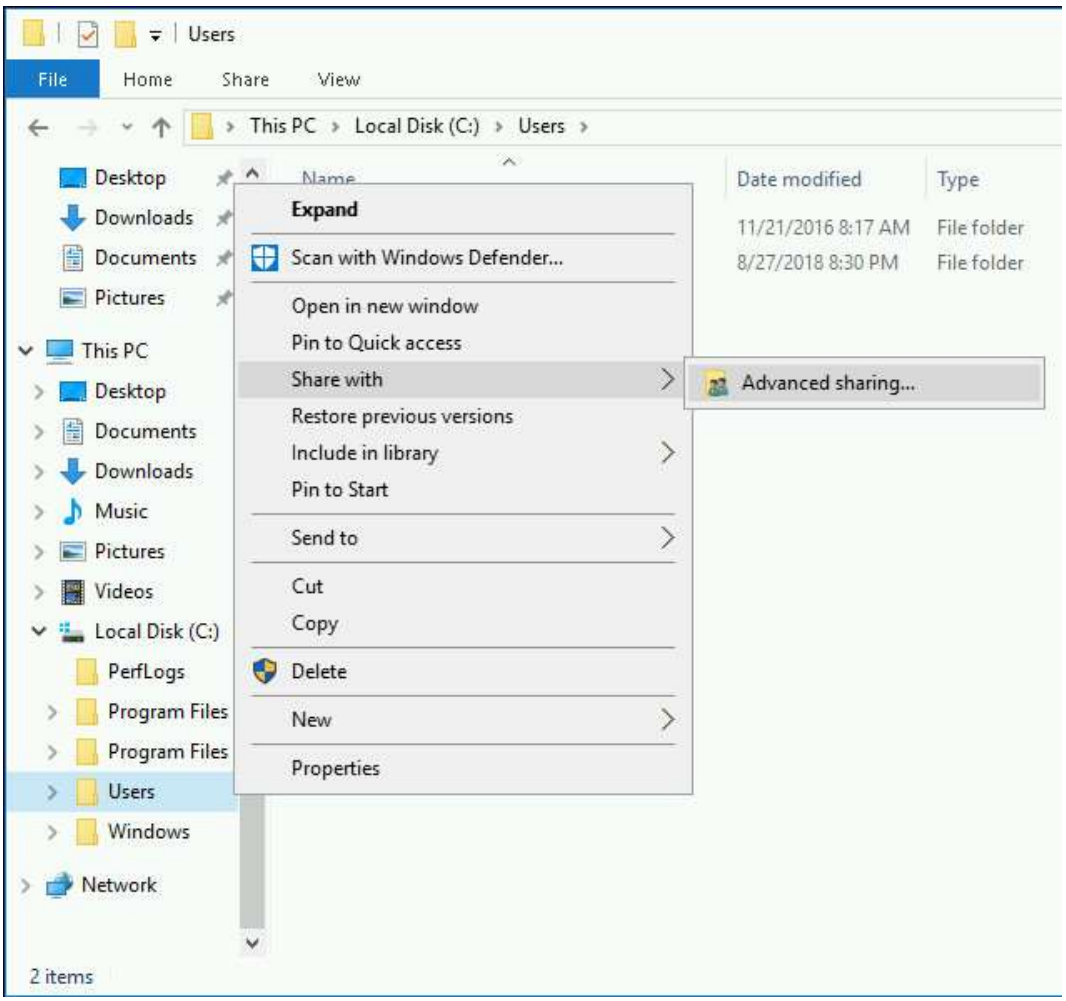
Then click on **Add a user account** to get a prompt for creating the new user.



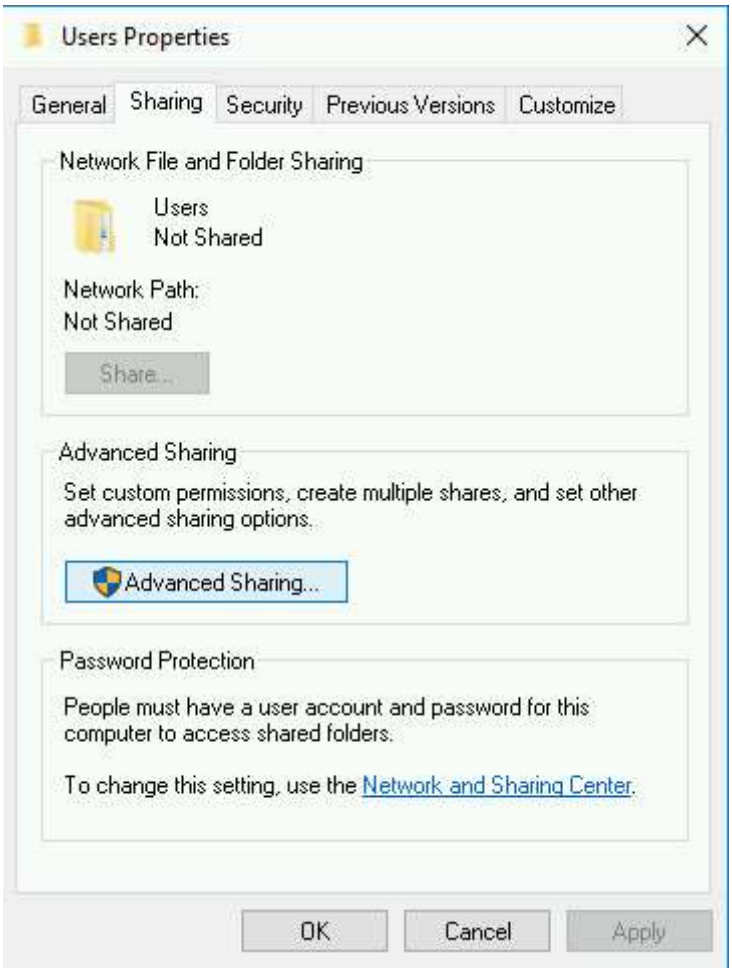
This window will allow you to create the new `backuppc` user, you also need to set a password for this account. The system will enforce that the password follows the default security policy. This means that it needs to have a combination of lower and upper case letters as well as a combination of letters and numbers or symbols.

Note: Note down the password which you have created. This will be further used in the lab.
Once we have created the user, we need to share the folder that we want to get backed up. In this scenario the folder that we want to backup is the `Users` folder.

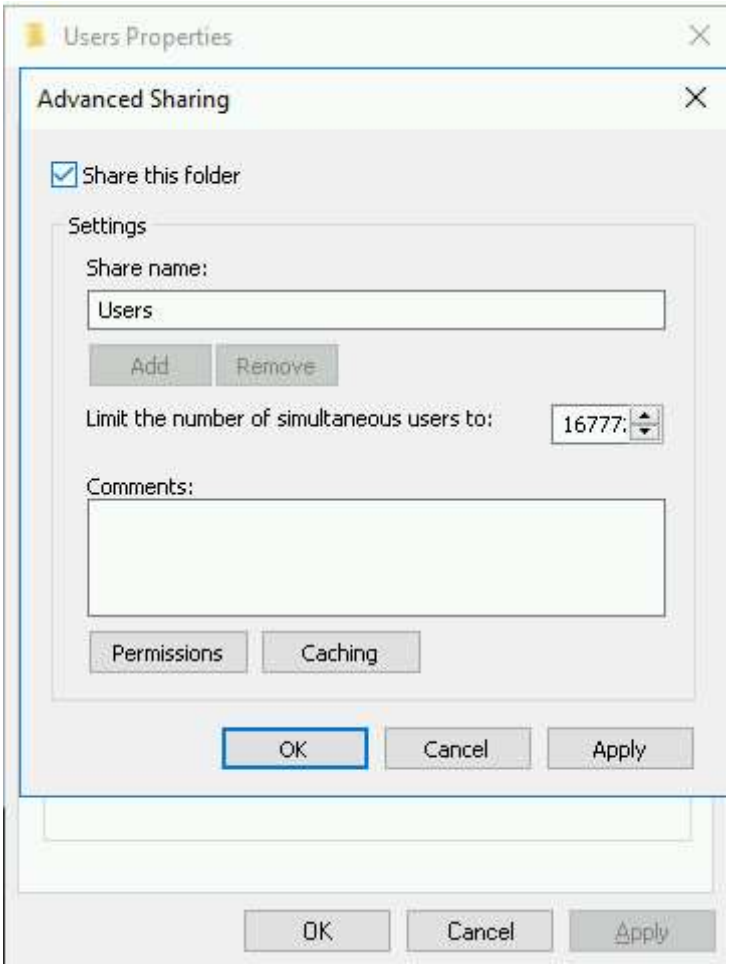
To share it, open the File Explorer, navigate to the `C:\Users` folder, right click on it and then click on **Share with > Advanced sharing...**



This will open the Properties window in the **Sharing** tab.

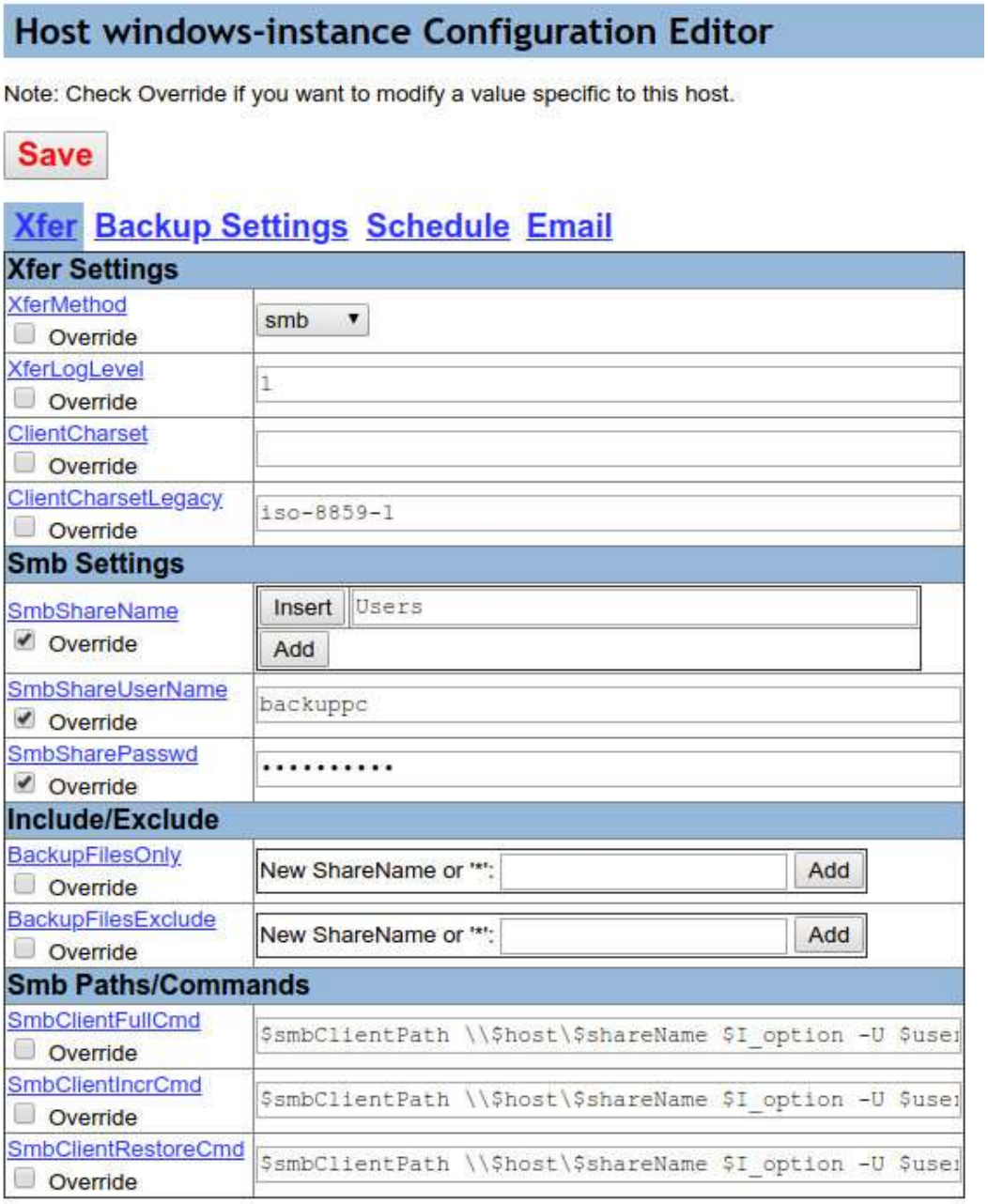


Click the **Advanced Sharing** button. This will open another window that will let you share the folder by selecting the "Share this folder" checkbox.



Press **Apply**, then **OK** on both windows to confirm that you want to share this folder. That's it for the windows configuration. Go back to your BackupPc administrative web interface to configure the host.

As you've done before, you first need to click on **Edit Hosts**, add an entry for `windows-instance` and save. After doing that, click on **Host Summary** and then click on `windows-instance`. From the home page, click on **Edit Config** to edit the configuration and select the **Xfer** tab.



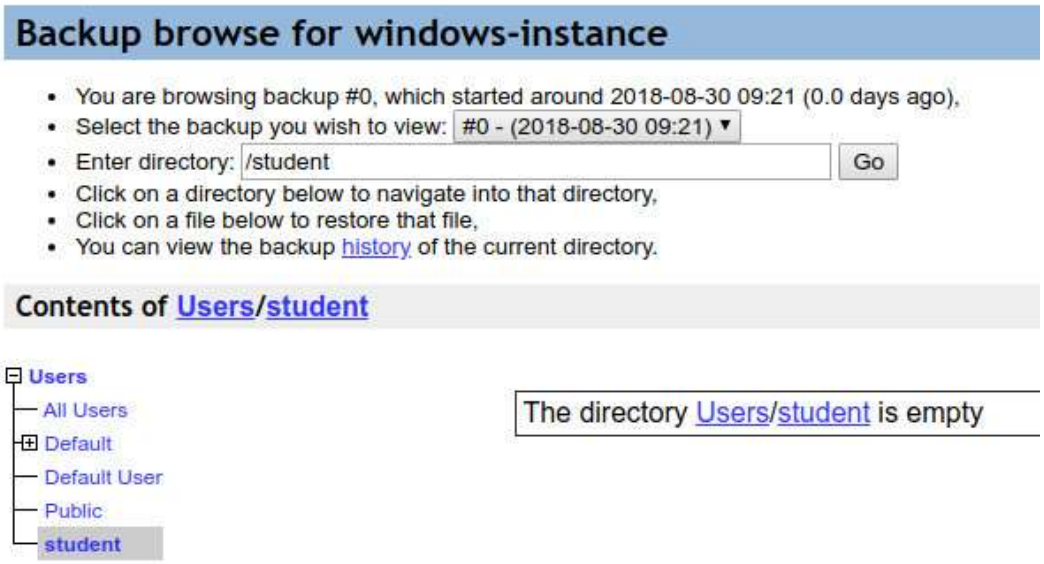
In this case, we are going to use `smb` as the **XferMethod**. This is also known as *samba* and it is the name of the protocol used to interact with shared Windows folders.

We need to configure the **SmbShareName** to be the folder that we want to backup (`Users`) and **SmbShareUserName** and **SmbSharePassword** to be the username and password that we've created on the Windows instance.

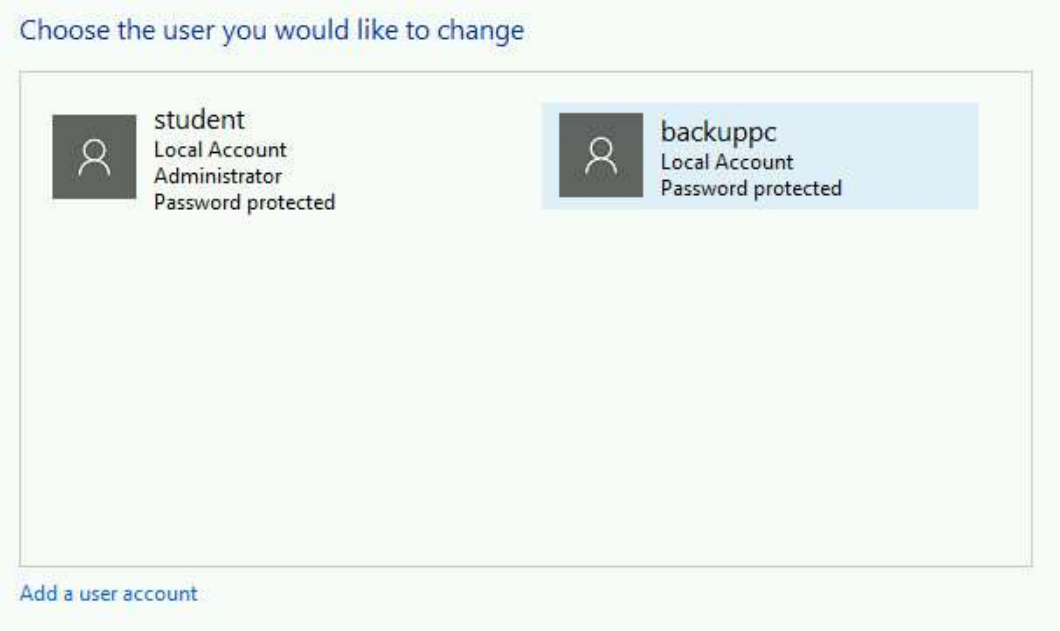
Once you've set that, click **Save** and go back to the home page for `windows-instance` and click on the **Start Full Backup** button to start a full backup. It will take a few moments to complete. Once it's done you should be able to browse the data that has been backed up, as done with the two other backups.

If you navigate to the directory corresponding to the `student` user, you'll get a message saying that the directory is empty. This is because the `backuppc` user that

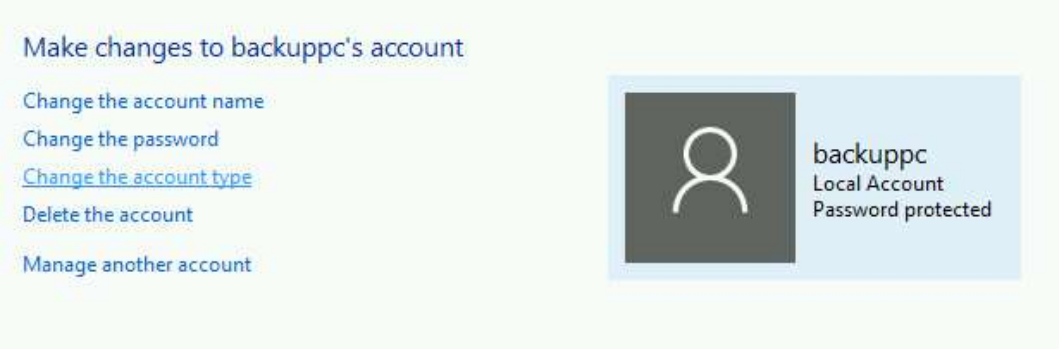
we created does not have permissions to see those files.



In order to fix that, we will change the account type of the `backuppc` user to an Administrator account. Go back to your windows session and in the **User Accounts** application select the `backuppc` user.



This lets us modify the configuration for the account we created.



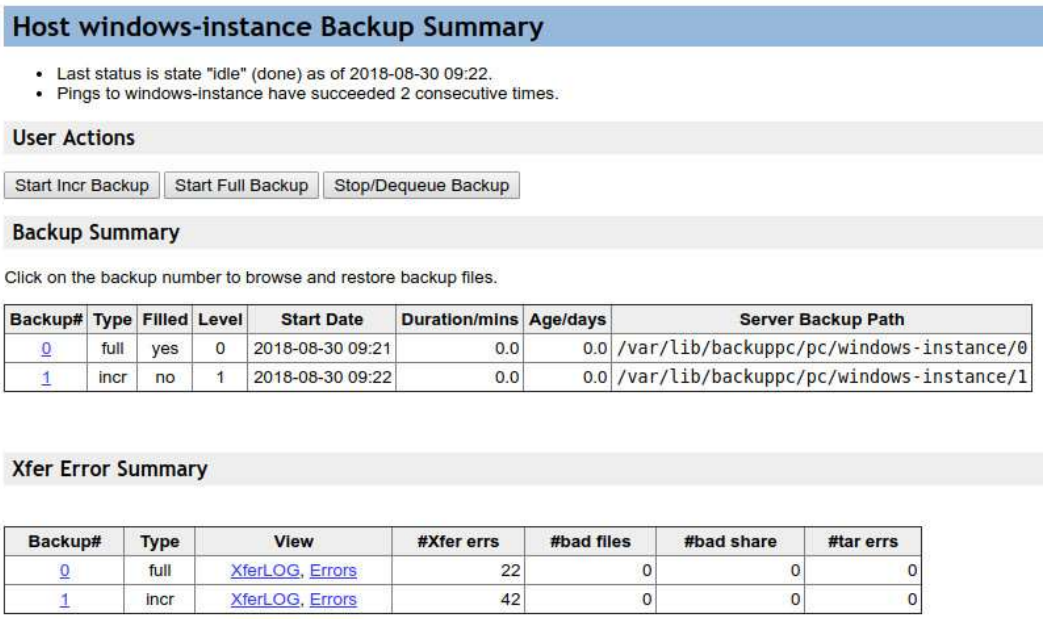
In this case, we want to **Change the account type**.

We want to change the account to Administrator. You can do this by selecting the **Administrator** option and then clicking on the **Change Account Type** button.

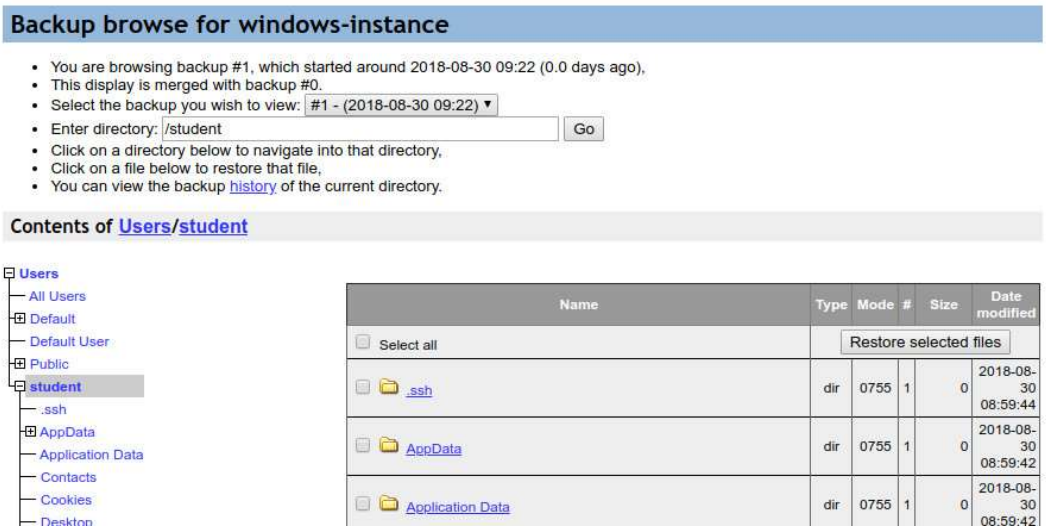
Now that we've made our `backuppc` user an administrator of the machine, let's retry the backup.

Go back to the **windows-instance Home** page in the BackupPc web interface. Press the **Start Incr Backup** button, this will start an incremental backup. This means that it won't download or store any files that it has already downloaded and stored. It will instead process only the differences between the previous backup and this one.

Once you've confirmed the incremental backup prompt, go back to the home page and you will see that there is a new entry for it.



Click on the **1** to browse that new backup. If you navigate to the student directory, you'll see that this time it contains all the files.



Success!

Click *Check my progress* to verify the objective.

Backup windows-instance Users directory

Conclusion

Wow! You've successfully configured a backup server, performed full and incremental backups, fixed permission errors, verified that the backups work correctly by downloading files and by performing restores, set up a backup server to backup both locally stored configuration files as well as user directories on remote Linux and Windows machines. Along the way, you've learned about managing the sudo configuration, exchanging SSH keys, connecting to shared folders on Windows machines, and more.

This has been a long journey, but you've put into practice many of your newly acquired system administrator skills and learned new tools that will help you in real-world scenarios.

Congratulations! This is quite an achievement!

End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you've used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.