

# Backup Strategies

---

Sumner Evans

October 22, 2020

Mines Linux Users Group

# Backup Principles

---

# Why backup? I

**Computers were a mistake.** But the bigger mistake was to give humans control over the computers.

Sometimes certain humans may write a program along the lines of:

```
with open("~/awesome", "w+") as f:  
    f.writeline("Awesome program\n")
```

which doesn't do what you expect because by default Python doesn't expand ~ by default meaning this creates a directory named ~ in your working directory.

Naturally, to delete this directory, you would run `rm -rf ~`, right?

(I may or may not have first-hand experience with this situation.)

# Why backup? I

**Computers were a mistake.** But the bigger mistake was to give humans control over the computers.

Sometimes certain humans may write a program along the lines of:

```
with open("~/awesome", "w+") as f:  
    f.writeline("Awesome program\n")
```

which doesn't do what you expect because by default Python doesn't expand ~ by default meaning this creates a directory named ~ in your working directory.

Naturally, to delete this directory, you would run `rm -rf ~`, right?

(I may or may not have first-hand experience with this situation.)

# Why backup? I

**Computers were a mistake.** But the bigger mistake was to give humans control over the computers.

Sometimes certain humans may write a program along the lines of:

```
with open("~/awesome", "w+") as f:  
    f.writeline("Awesome program\n")
```

which doesn't do what you expect because by default Python doesn't expand ~ by default meaning this creates a directory named ~ in your working directory.

Naturally, to delete this directory, you would run `rm -rf ~`, right?

(I may or may not have first-hand experience with this situation.)

# Why backup? I

**Computers were a mistake.** But the bigger mistake was to give humans control over the computers.

Sometimes certain humans may write a program along the lines of:

```
with open("~/awesome", "w+") as f:  
    f.writeline("Awesome program\n")
```

which doesn't do what you expect because by default Python doesn't expand ~ by default meaning this creates a directory named ~ in your working directory.

Naturally, to delete this directory, you would run `rm -rf ~`, right?

(I may or may not have first-hand experience with this situation.)

# Why backup? I

**Computers were a mistake.** But the bigger mistake was to give humans control over the computers.

Sometimes certain humans may write a program along the lines of:

```
with open("~/awesome", "w+") as f:  
    f.writeline("Awesome program\n")
```

which doesn't do what you expect because by default Python doesn't expand ~ by default meaning this creates a directory named ~ in your working directory.

Naturally, to delete this directory, you would run `rm -rf ~`, right?

(I may or may not have first-hand experience with this situation.)

# Why backup? I

**Computers were a mistake.** But the bigger mistake was to give humans control over the computers.

Sometimes certain humans may write a program along the lines of:

```
with open("~/awesome", "w+") as f:  
    f.writeline("Awesome program\n")
```

which doesn't do what you expect because by default Python doesn't expand ~ by default meaning this creates a directory named ~ in your working directory.

Naturally, to delete this directory, you would run `rm -rf ~`, right?

(I may or may not have first-hand experience with this situation.)



## Why backup? II

### **Ransomware** protection.

If a malicious actor manages to encrypt a bunch of the files on your filesystem and demands money to get the key, you can just restore to a previous backup with minimal loss of productivity.

## Why backup? II

**Ransomware** protection.

If a malicious actor manages to encrypt a bunch of the files on your filesystem and demands money to get the key, you can just restore to a previous backup with minimal loss of productivity.

# Why not backup?

If you don't have any data that is important to you, and like setting up your computer over and over again, then you don't need a backup.

I'd bet that you have *something* that you want to backup.

# Why not backup?

If you don't have any data that is important to you, and like setting up your computer over and over again, then you don't need a backup.

I'd bet that you have *something* that you want to backup.

# Why not backup?

If you don't have any data that is important to you, and like setting up your computer over and over again, then you don't need a backup.

I'd bet that you have *something* that you want to backup.

# Why not backup?

If you don't have any data that is important to you, and like setting up your computer over and over again, then you don't need a backup.

I'd bet that you have *something* that you want to backup.

# Don't backup everything!

Backups can get bloated if you include too many unimportant files!

- A very small number of your dotfiles are actually useful to be backed up.
- Likewise only a few files in `/etc` actually matter.
- `/var` sometimes contains things that are worth backing up.
- On Windows, it's pointless to backup `C:\\Program Files`.

# Don't backup everything!

Backups can get bloated if you include too many unimportant files!

- A very small number of your dotfiles are actually useful to be backed up.
- Likewise only a few files in `/etc` actually matter.
- `/var` sometimes contains things that are worth backing up.
- On Windows, it's pointless to backup `C:\\Program Files`.



# Don't backup everything!

Backups can get bloated if you include too many unimportant files!

- A very small number of your dotfiles are actually useful to be backed up.
- Likewise only a few files in `/etc` actually matter.
- `/var` sometimes contains things that are worth backing up.
- On Windows, it's pointless to backup `C:\\Program Files`.

# Don't backup everything!

Backups can get bloated if you include too many unimportant files!

- A very small number of your dotfiles are actually useful to be backed up.
- Likewise only a few files in `/etc` actually matter.
- `/var` sometimes contains things that are worth backing up.
- On Windows, it's pointless to backup `C:\\Program Files`.

# The 3-2-1 rule

The **3-2-1 backup rule** states that you should:

Keep at least **three** copies of your data on at least **two** different storage media and store at least **one** of the copies off-site.

This may sound daunting, but keep in mind that **any backup is better than no backup!** You have to start somewhere.

# The 3-2-1 rule

The **3-2-1 backup rule** states that you should:

Keep at least **three** copies of your data on at least **two** different storage media and store at least **one** of the copies off-site.

This may sound daunting, but keep in mind that **any backup is better than no backup!** You have to start somewhere.

# The 3-2-1 rule

The **3-2-1 backup rule** states that you should:

Keep at least **three** copies of your data on at least **two** different storage media and store at least **one** of the copies off-site.

This may sound daunting, but keep in mind that **any backup is better than no backup!** You have to start somewhere.

# The 3-2-1 rule

The **3-2-1 backup rule** states that you should:

Keep at least **three** copies of your data on at least **two** different storage media and store at least **one** of the copies off-site.

This may sound daunting, but keep in mind that **any backup is better than no backup!** You have to start somewhere.

## A few other principles

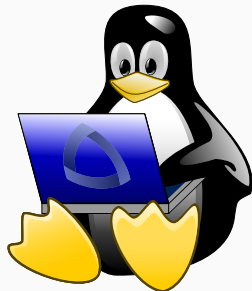
**The best backups are automatic**, because otherwise you'll always default to “oh, I can do that later”.

**Tailor your backups to the data your are backing up.** For example, don't just backup all of the files that your database uses, rather export your database periodically and backup that export.

# Copyright Notice

This presentation was from the **Mines Linux Users Group**. A mostly-complete archive of our presentations can be found online at <https://lug.mines.edu>.

Individual authors may have certain copyright or licensing restrictions on their presentations. Please be certain to contact the original author to obtain permission to reuse or distribute these slides.



Colorado School of Mines  
Linux Users Group