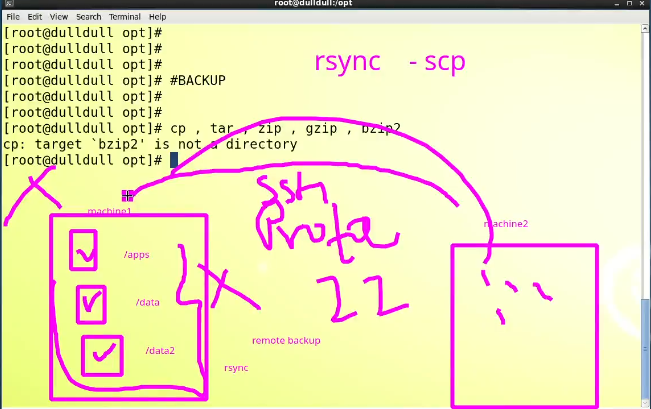
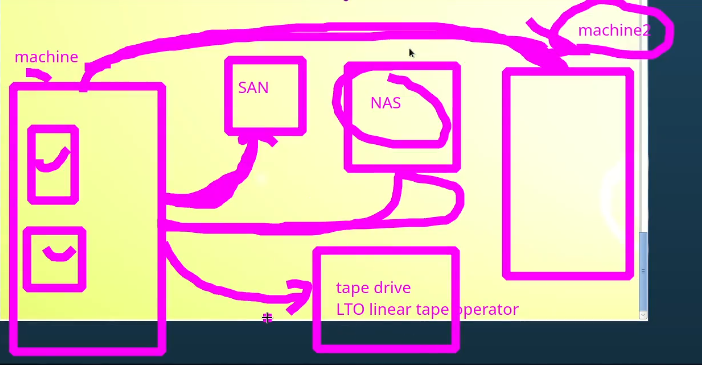
Lecture 14

Rsync (backup)

Backup tools

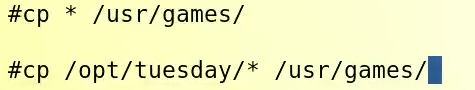
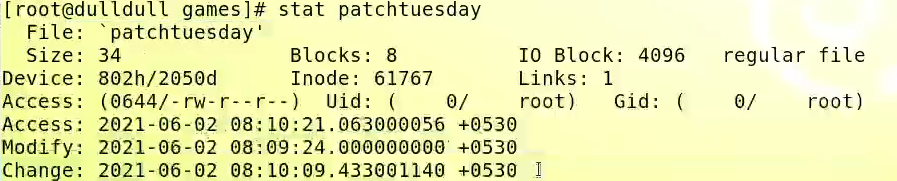
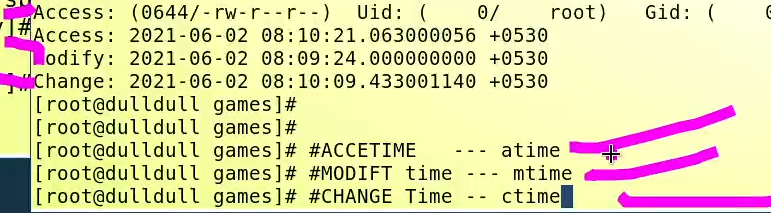
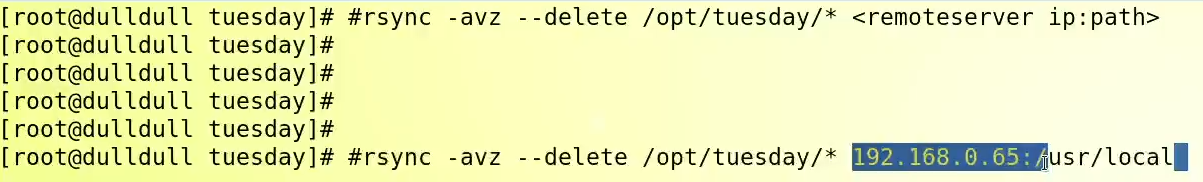
**Rsync**

* From one machine to another machine 🡪 proper backup
* Backup tool
* Rsync & scp
* 
* Types of backup (interview Question)
  + Full backup
  + Incremental backup
    - As any file is modified
  + Differential backup
    - A differential backup is a type of backup that only backs up the changes made to a file or set of files since the last full backup. This means that only new or modified data is backed up, rather than all data. Differential backups are typically faster and take up less storage space than full backups, but they do require that a full backup be performed first in order to be effective. When restoring data from a differential backup, the most recent full backup and the most recent differential backup must be used together in order to restore the data to its most recent state.
  + Cumulative backup
    - A cumulative backup is a type of incremental backup that captures all changes made to files since the last cumulative backup and requires all previous cumulative backups and the most recent cumulative backup to restore data to its most recent state.
* Rsync can be used to have backups with compression.
  + **$ rsync | tar or $ rsync | zip**
* Data taking options. (Interview question)
* 

Tip:

If you want to check if a directory is available and want to be in it



* $ cd ../<directory\_name>/
* “cp” command
* To copy contents of a directory to another directory (only files)
* 
* 2nd command is a great option to copy all the contents of a specific directory
* $ cp -rv <source\_path> <destination\_path>
* For example,
* $ **cp -rv /opt/Tuesday/\* /usr/games**/ 🡪 this commands copies all the contents from source to destination including directories
* **Every time this command runs it overwrites the contents in destination. And it’s not a good practice** specially if only one file is added in source, and it is to be backed up to destination directory
* That’s why “rsync” is a best tool which do things smartly.
* **$ rsync -rv <source>\* <destination>**
* -parv 🡪 **p** flag preserve permission (**it means the ownership and permissions of the files remains with original owner**, the person who is running the command or the user may not become owner of the files), **a** archive, **r** recursive **v** verbose
* -avz 🡪 **a** flag archives, **v** verbose **z** (gzip) compresses
* “rsync” can not copy files from destination to source. If there is an extra file in destination, “rsync” can delete it by just adding a flag “- -delete”
* **15 examples of “rsync” usage,**
  1. Backup a local directory to another location on the same machine: **rsync -avz /original/path /backup/path**
  2. Backup a local directory to another location on the same machine and delete files from the backup that are no longer in the original directory: **rsync -avz --delete /original/path /backup/path**
  3. Backup a local directory to a external drive: **rsync -avz /original/path /mnt/external\_drive/backup**
  4. Schedule a daily backup of a local directory using cron: **0 0 \* \* \* rsync -avz /original/path /backup/path**
  5. Backup a local directory and exclude a specific file type: **rsync -avz --exclude '\*.log' /original/path /backup/path**
  6. Backup a local directory and only include files larger than 100MB: **rsync -avz --min-size='100M' /original/path /backup/path**
  7. Backup a local directory and only include files modified in the last 24 hours: **rsync -avz --modify-window=24 /original/path /backup/path**
  8. Backup a local directory and include hidden files and directories: **rsync -avz --include='\*/' --include='.\*' --exclude='\*' /original/path /backup/path**
  9. Backup a local directory and use a specific ssh key: **rsync -avz -e "ssh -i /path/to/key" /original/path /backup/path**
  10. Backup a local directory to a remote server with compression: **rsync -avz -e "ssh -c arcfour" /original/path user@remote:/backup/path**
  11. Schedule a weekly backup of a local directory using cron: **0 0 \* \* 7 rsync -avz /original/path /backup/path**
  12. Backup a local directory and use a specific ssh key with a specific username: **rsync -avz -e "ssh -l username -i /path/to/key" /original/path /backup/path**
  13. Backup a local directory and also preserve permissions, ownership, and timestamps: **rsync -avzpogt /original/path /backup/path**
  14. Backup a local directory with a specific bandwidth limit: **rsync -avz --bwlimit=1000 /original/path /backup/path**
  15. Backup a local directory and split large files into smaller chunks: **rsync -avz --chunk-size=64M /original/path /backup/path**
* **Stat command**
* ****
* **Interview question**
* ****
* **Find atime, ctime, mtime**
* atime 🡪 if anyone opens the file its called access time
* mtime 🡪 modify time , if file is accessed and modified, both atime and m time changes
* ctime 🡪 if ownership and permission of the file is changes, atime is also changed
* it is used with “find” command.
* **how “rsync” works**
  + it compares modify time of source and destination and if any change is detected, it will copy the incremental data to destination.
* **Remote sync**
* 
* Chart, text

  Description automatically generated
* And remote destination command would be like,
* 
* In industry – there would be backup application or backup server.
  + Backup polices would be in place.
  + Veritas NetBackup 🡪 solution
  + Windows NT Backup
  + TS
  + VEEAM
  + Dell Networker
  + EMC Networker
* There r so many opensource backup applications, AMANADA , ZAMANDA, BACULA and BACKUPC.