**DevOps Domain**

**- Linux Training-**

# Table of contents

Contents

[1. Table of contents 2](#_Toc67404044)

[Contents 2](#_Toc67404045)

[2. Document history 2](#_Toc67404046)

[3. Task description 3](#_Toc67404047)

[3.1 Lab environment: VM access 3](#_Toc67404048)

[3.2 Lab environment: Files/directories permissions 3](#_Toc67404049)

[3.3 Lab environment: Files/directories manipulation 3](#_Toc67404050)

[3.4 Lab environment: commands find, grep, cut, pipe 4](#_Toc67404051)

[3.5 Lab environment: environmental variables 4](#_Toc67404052)

[3.6 Lab environment: sed, awk (advanced) 5](#_Toc67404053)

[3.7 Lab environment: Cron job 5](#_Toc67404054)

[3.8 Lab environment: Bash scripting 6](#_Toc67404055)

[3.8.1 Script #1 6](#_Toc67404056)

[3.8.2 Script #2 6](#_Toc67404057)

[3.8.3 Script #3 7](#_Toc67404058)

[4. Documentation 7](#_Toc67404059)

# Document history

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Version | Comments |
|  | 04.03.2021. | 1.0 | Bash scripting chapter missing. |
|  | 08.03.2021. | 1.1 | Script #1 |
|  | 12.03.2021. | 1.2 | Script #2 |
|  | 16.03.2021. | 1.3 | Script #3 - A |
|  | 25.03.2021. | 1.4 | Script #3 - B |
|  | 29.03.2021 | 1.5 | Change the titles, change the nsmiljanic with <username> |

# Task description

## Lab environment: VM access

VM IP address: 192.168.10.200

Credentials:

Username: <username>

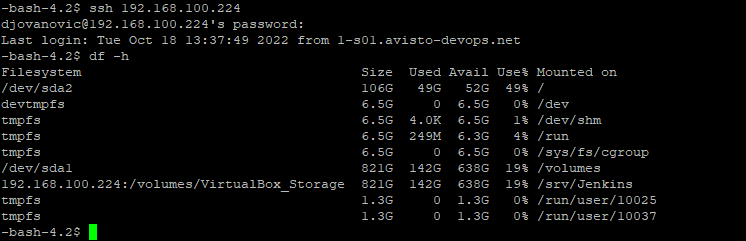
Password: xxxxxx

* Check VM access rights:
  + Connect to Avisto Eastern Europe’s LAN using VPN [Forty client] (mandatory step only for users working from home office).
  + Open SuperPuTTY terminal (or any other terminal emulator of your choice) and connect to VM with provided credentials.
* Exercise:
  + Create SSH key pair, add public SSH key (<username> user) and connect to VM using SSH Private key.
    - SSH key pair is created by default when connecting via PuTTY
  + With which group(s) is your user account associated with?

id djovanovic

uid=10037(djovanovic) gid=10000(avisto-devops) groups=10000(avisto-devops)

* + Do you have root access? If not, how can you get it?
    - No, by default user with id 0 has root access. When I tried sudo, access is denied.
    - Permissions can be changed if user is added to /etc/sudoers, where list of root permissions users is stored. This can be managed by admin.
  + Add alias ll to “ls -la” command. Can you use this alias next time you login to VM? If not, how can you fix it?
    - Alias ll=”ls -la”
    - If we want to save alias, we should crate ~/.bashrc file and store aliases in it. When we use command source ~/.bashrc we now have aliases written there
  + Connect to VM 192.168.100.224 with your user account from 192.168.10.200 machine. Check the host name and size of root partition.



\* For each task, document used commands and output (if applicable).

## Lab environment: Files/directories permissions

* Create directory *<username>\_lab* under your home directory. Create two files named date1.txt and date2.txt having current date as content. Go under directory *<username>\_lab* and type *ls -l*:
  + From left to right, explain each field.
    - -rw- r-- r-- - Permissions
    - 1 – number of linked hard links
    - djovanovic – owner of a file
    - avisto-devops – group that owns a file
    - Oct 18 12:03 – date and time
    - date1.txt – filename
  + Explain date1.txt file permissions (octal representation).
    - Octal representation is 644 (110 100 100)
  + Copy sleep.sh script from *~/tgengo\_lab* directory under *<username>\_lab* directory and run it. What is the output?
    - sh ~/djovanovic\_lab/sleep.sh
    - “Wait for 5 seconds”
    - “Completed”
  + Go under *~/tgengo\_lab/scripts/* and run *date\_parse.sh* script. What is the output? Explain the result. Try to overcome problem.
    - sh date\_parse.sh
    - Tue Oct 18 15:40:25 CEST 2022
    - Current Date is: 18-10-2022
    - Current Time is: 15:40:25

\* For each task, document used commands and output (if applicable).

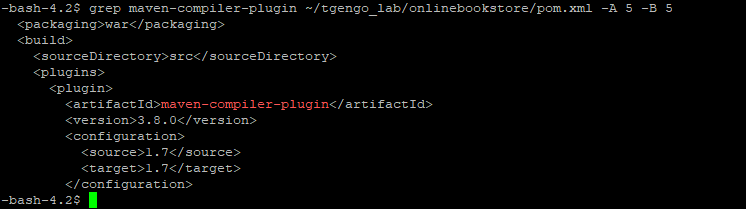
## Lab environment: Files/directories manipulation

* Create *project* directory under *$HOME/<username>\_lab* directory and nine empty files (house1 to house9) under it using for loop.
  + for I in {1..9} do
  + touch house$i
  + done
* Create some empty files: *$HOME/<username>\_lab/projects/houses/bungalow.txt*, *$HOME/<username>\_lab/projects/houses/doors/bifold.txt*, *$HOME/<username>\_lab/projects/outdoors/vegetation/landscape.txt*.
  + Touch ~/djovanovic\_lab/houses/bungalow.txt
  + touch ~/djovanovic\_lab/houses/bifold.txt
  + touch ~/djovanovic\_lab/houses/landscape.txt
* Copy the files house1 and house5 to the *$HOME/ <username>\_lab/projects/houses/* directory keeping same file permissions.
  + Cp house{1..5} ~/djovanovic\_lab/projects/houses/
* Recursively copy the */usr/share/doc/initscripts\** directory to the *$HOME/<username>\_lab/projects/* directory.
  + Cp -r /usr/share/doc/initscripts\* ~/djovanovic\_lab/projects/
* Recursively list the contents of the *$HOME/<username>\_lab/projects/* directory. Pipe the output to the *less* command so you can page through the output.
  + Ls -r ~/djovanovic\_lab/projects/ | less
* Remove the files house6, house7, and house8 without being prompted.
  + Rm house{6..8}
* Move house3 and house4 to the *$HOME/<username>\_lab /projects/houses/doors* directory.
  + Mv house{3..4} ~/djovanovic\_lab/projects/houses/doors
* Remove the *$HOME/<username>\_lab /projects/houses/doors* directory and its contents.
  + Rm -r ~/djovanovic\_lab/projects/houses/doors
* Change the permissions on the *$HOME//<username>\_lab /projects/house2* file so it can be read and written to by the user who owns the file, only read by the group, and have no permission for others.
  + Chmod 640 ~/djovanovic\_lab/projects/house2
* Recursively change the permissions of the *$HOME <username>\_lab /projects/* directory so that nobody has write permission to any files or directory beneath that point in the file system.
  + chmod -R a-w $HOME/djovanovic\_lab/projects/
* Copy create\_dir.sh script, using *scp* command, from /tmp/<username>\_224 directory on 192.168.100.224 server under *~/<username>\_lab* directory on 192.168.10.200 VM and run it. What is the result? Provide, at least one more, solution to copy script from one server to another.
  + Scp [djovanovic@192.168.100.224](mailto:djovanovic@192.168.100.224):/tmp/ntomic\_224/create\_dir.sh [djovanovic@192.168.10.200](mailto:djovanovic@192.168.10.200):djovanovic\_lab/
  + create\_dir.sh 100% 306 48.6KB/s 00:00
  + Connection to 192.168.100.224 closed.
  + Alternative to this is SFTP

\* For each task, document used commands and output (if applicable).

## Lab environment: commands find, grep, cut, pipe

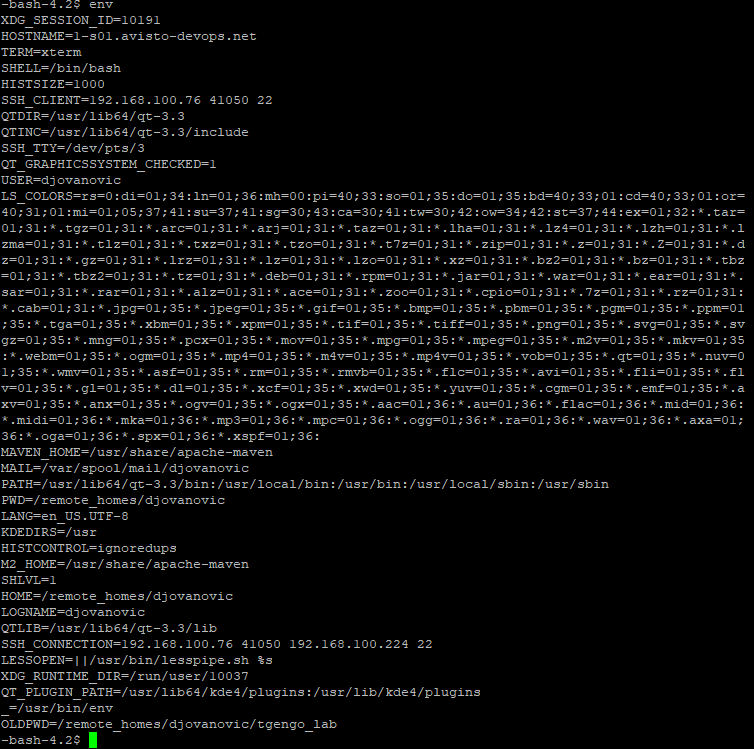
* Create a *~/<username>\_lab*/FILES directory. Find all files under the /usr/share/backgrounds/ directory that are more than 5MB and less than 10MB and copy them to the *~/<username>\_lab*/FILES directory.
  + Mkdir ~/djovanovic/FILES
  + find /usr/share/backgrounds/ -type f -size +5M -size -10M -exec cp "{}" ~/djovanovic\_lab/FILES/ \;
* Find every file in the ~/<username>\_lab/FILES directory and make a backup copy of each file in the same directory. Use each file’s existing name and just append .mybackup to create each backup file.
  + find ~/djovanovic\_lab/FILES/ -type f -exec cp "{}" ~/djovanovic\_lab/FILES/{}.mybackup \’
* Find every file with .java extension under *~/tgengo\_lab/onlinebookstore* directory and create tar archive named javaFiles.tar under *~/<username>\_lab* directory.
  + Find ~/djovanovic\_lab/FILES/ -name “\*.java” -exec tar -rvf ~/djovanovic\_lab/javaFiles.tar
  + -r append to end of archive
  + -v verbose
  + -f file
* Run *pwd* command under *~/tgengo\_lab/onlinebookstore/* directory and extract *<username>* string from the result using *cut* command. Is there any other way to extract it?
  + pwd | cut -d "/" -f 3
  + we can also do this with awk
  + pwd | awk -F'/' '{print $3}'
    - F – field separator
* How many files under *~/tgengo\_lab/onlinebookstore/* have word “servlets” in the content?
  + grep -r servlets ~/tgengo\_lab/onlinebookstore/ |wc -w
  + 38
* Print 5 lines before and after matched “maven-compiler-plugin” key word in *~/tgengo\_lab/onlinebookstore/pom.xml* file.



\* For each task, document used commands and output (if applicable).

## Lab environment: environmental variables

* What do you get as a result running *env* command?



* Create env variable ENV\_VAR and associate value “This is test variable” to it. How can you check the variable value?
  + ENV\_VAR="This is a test variable"
  + echo $ENV\_VAR
* Do you have ENV\_VAR env variable available once you logout/login to your VM? Explain.
  + No, temporary variables are cleared after restart
* Create env variable COMMAND\_VAR as result of command that extract parent directory of your current directory path. Can you call this variable in the script and get result?
  + filepath=$(pwd)
  + parentdir=${filepath%/\*}
  + echo $parentdir

\* For each task, document used commands and output (if applicable).

## Lab environment: sed, awk (advanced)

* *sed*: File *~/tgengo\_lab/phonebook*
  + Change the name Jon to Jonathan.
    - sed -i 's/Jon/Jonathan/g' ~/djovanovic\_lab/phonebook
    - before that I copied file to my folder, because of no permissions in tgengo\_lab folder
  + Delete the first three lines.
    - sed -i '1,3d' ~/djovanovic\_lab/phonebook
  + Print lines 5 through 10.
    - sed -n '5,10'p phonebook
  + Delete lines containing Lane.
    - sed -i '/Lane/d' phonebook
  + Print all lines where the birthdays are in November or December.
    - sed -n -E '/11\/|12\//'p phonebook
  + Append three asterisks to the end of lines starting with Fred.
    - sed -i '/^Igor/ s/$/\*\*\*/' phonebook
    - Fred smo obrisali pa sad je Igor umesto
  + Replace the line containing Jose with JOSE HAS RETIRED.
    - sed -i 's/^Jose.\*/JOSE HAS RETIRED/g' phonebook
  + Change Popeye 's birthday to 11/14/46. Assume you do not know Popeye's original birthday. Use a regular expression to search for it.
    - sed -i '\|^Popeye|s|:[0-9]\*\/.\*\/..|:11/14/46|' phonebook
  + Delete all blank lines.
    - Sed -i '/^$/d' phonebook
  + Write a sed script that will:
    - Insert above the first line the title PERSONNEL FILE.
      * sed -i '1i\PERSONNEL FILE' phonebook
    - Remove the salaries ending in 500.

Print the contents of the file with the last names and first names reversed.

* + - * sed -i 's/:[0-9]\*500$/:/g' phonebook
      * sed -i 's/^\([^ ]\*\) \([^:]\*\)/\2 \1/' phonebook
    - Append at the end of the file THE END.
      * sed -i '$a\THE END' phonebook
  + touch sed-script.sh
    - vi sed-script.sh and input these commands
    - sh sed-script.sh
* *Awk*: File *~/tgengo\_lab/database* (The database contains the names, phone numbers, and money contributions to the party campaign for the past three months.)*:*
  + Print all the phone numbers.
    - -bash-4.2$ cat database | awk -F ':' '{print $2}'
  + Print Dan 's phone number.
    - grep Dan database | awk -F ':' '{print $2}'
  + Print Susan 's name and phone number.
    - grep Susan database | awk -F ':' '{print $1" "$2}'
  + Print all last names beginning with D.
    - grep " D" database | awk -F ' ' '{print $2}' | awk -F ':' '{print $1}'
  + Print all first names beginning with either a C or E.
    - grep -E '^C|^E' database | awk -F ' ' '{print $1}'
  + Print all first names containing only four characters.
    - grep '^.... ' database | awk -F ' ' '{print $1}'
  + Print the first names of all those in the 916area code.
    - grep \(916\) database | awk -F ' ' '{print $1}'
  + Print Mike 's campaign contributions. Each value should be printed with a leading dollar sign e.g., $250 $100 $175.
    - grep Mike database | awk -F ':' '{print "$"$3 " $"$4 " $"$5}'
  + Print last names followed by a comma and the first name.
    - grep '[^&]' database | awk -F '[^A-z]' '{print $2 ", " $1}'
  + Write an awk script called facts that:
    - Prints full names and phone numbers for the Savages.
    - Prints Chet 's contributions.
    - Prints all those who contributed $250 the first month.
      * grep Savage database | awk -F ':' '{print $2 " " $2}'
      * echo
      * echo "Chet's contributions: "
      * grep Chet database | awk -F ':' '{print "$"$3 " $"$4 " $"$5}'
      * echo
      * echo "Contributed $250 in the first month: "
      * grep 250: database | awk -F ':' '{print $1}'

\* For each task, document used commands and output (if applicable).

## Lab environment: Cron job

* Explain Linux *cron* utility.
  + It is used for scheduling tasks in Linux
* Run command to list all *cron jobs*.
  + crontab -l
    - no crontab for djovanovic
* Create a *cron job* that executes *sleep.sh* script every day at 1:15 PM.
  + Crontab -e
  + 15 13 \* \* \* ~/djovanovic\_lab/sleep.sh
* Create a *cron job* that executes date\_parse.sh script every hour.
  + 00 \* \* \* \* ~/djovanovic\_lab/date\_parse.sh
* Create a *cron job* to list all "\*.sh" files in the *~/<username>\_lab* directory and redirect to the following file "~/ *<username>\_lab* /allsh.txt
  + Set crontab to run at: 10 mins from your current time.
    - Touch list\_and\_redirect.sh
    - vi list\_and\_redirect.sh
    - ls ~/djovanovic\_lab/ | grep .sh > allsh.txt
    - crontab -e
    - \*/10 \* \* \* \* ~/djovanovic\_lab/list\_and\_redirect.sh
* Create a cron job to remove all empty log files in your $HOME directory;
  + Set crontab to run at: every mon-frid @ 12 am for the next 4 months.
    - #! /bin/bash
    - find ~/ -type f -name \*.log -empty -delete
    - Crontab -e
    - 0 12 \* 10,11,12,1 1-5 ~/djovanovic\_lab/find\_and\_delete.sh
* How can you check if the script is executed at specified time?
  + Grep <ime\_skripte> /var/log/cron – medjutim nemamo permission za citanje ovog fajla

\* Contact Tamara Gengo to provide you root privileges, if needed.

## Lab environment: Bash scripting

### Script #1

* Target language: Bash
* Target OS: Linux
  + A: Script name: archivedir -- Creates a compressed archive (tar) of the specified directory
    - Features:
      * Input parameter(s) – directory
      * What needs to be checked and user informed about (if applicable):
      * If there are no arguments or more than 1.
      * Check if directory exists.
      * Check if current directory is specified as input parameter.
      * Check if you have permissions to write compressed archive to current directory.
        + Check if directory size is bigger than 10 (size in blocks), warn user about it and ask if he wants to proceed.
      * Check if you have enough space on disk to save compressed file.
  + B: Script name: Choose adequate name for the script
    - Prerequisite: Install latest Git version
      * sudo yum install git
    - Features:
      * Ask user interactively for username, email, editor, if he wants to add alias, etc.
      * <https://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration>

### Script #2

* Target language: Bash
* Target OS: Linux
* Script name: setup.project – Create user workspace
* Prerequisite: Change ownership for ~/ tgengo\_lab/artifactory, ~/ tgengo\_lab/nexus and ~/ tgengo\_lab/docker-compose directories and its content from root:root to <username>:avisto-devops
* Features:
  + Interactively provide user option to choose environment:
    - 0 – artifactory
    - 1 – nexus
    - 2 – docker-compose
    - Abort operation: “.”
  + Ask user which name he wants to choose for his workspace (allowed only letters, small and capitals, numbers, underscore, and dash signs)
  + Create directory with chosen name and initialize it as git directory (*git init* command).
  + Ask user if he wants to stay on master branch or create new git branch.
  + If he wants to go with new branch, ask for the branch name and create it (git checkout -b <branch name> command).
  + Create symbolic links to all files from chosen environment till level 3. Exclude .git directory and files which names starts with “.” sign.
  + Inform user that workspace is created, where it is located and provide him some statistics about space left on machine, number of running processes and similar.

### Script #3

* Target language: Bash
* Target OS: Linux
  + A: Script name: reclaim-disk-space
    - Prerequisite: Install [Docker](https://docs.docker.com/engine/install/)
    - Features:
      * Implement help option (-h).
      * Implement dry-run with information what will be removed and ask user if he wants to proceed.
      * Remove exited containers (-c).
      * Remove unused images (-i).
      * Remove unused volumes (-v).
      * Remove all above (-all).
      * If you invoke script without arguments, print help.
      * Print statistics.
  + B: Script name: log-rotate
    - Prerequisite: Create dummy log files/directories under different paths (e.g., ~/app1/error1.log, ~/app1/error2.log, ~/app2/admin.log, etc.)
    - Features:
      * Script has one mandatory parameter – Configuration File Path (log-rotate.conf file) and one optional parameter – Directory where archive will be kept.
      * Log-rotate.conf file content example (it points to log files and directories):

#~/app1/subapp1/logs (# mark excludes the directory from log rotation)

~/app2/subapp2/logs

~/app1/error1.log

~/app2/subapp2/admin.log

* + - * Script should do log filesr rotation:
        + for log files in log-rotate.conf file create zip archive with date & timestamp in the name and remove the log file.
        + for directories in log-rotate.conf file, create zip archive with log files older than 5 hours (you can use any time suitable for easier script testing), having date & timestamp in the name. Remove archived log files.
      * Be careful about active log(s), as if it rotates an *active* log the application may not be able to write further log entries to the log file!

# Documentation

Use any available documentation on the Internet or contact Tamara Gengo for help.