1 Linux Admin

1.1 Resources

Books

1. Unix and Linux System Administration Handbook (Ordered)
2. The Practice of System and Network Administration

Communities

1. Superuser → `https://superuser.com/`
2. Server fault → `https://serverfault.com/`
3. Digital Ocean → `https://www.digitalocean.com/community/tutorials`

Sites

1. Ubuntu → `https://help.ubuntu.com/`
2. Tutorial Linux → `https://tutorialinux.com/`

Links

1. `https://www.slideshare.net/kavyasri790693/linux-admin-interview-questions`
2. `http://simplylinuxfaq.blogspot.in/p/linux-system-admin-interview-questions.html`
3. `https://github.com/kylejohnson/linux-sysadmin-interview-questions/blob/master/test.md`
4. `https://github.com/chassing/linux-sysadmin-interview-questions#hard`

1.2 Users, Passwords & Permissions

Users

1	Adding a user	useradd (single) → newusers (batch mode useradd)
2	Lock an Account	usermod -l ____
3	New password	passwd "username"
4	Default file permissions	Set UMASK in /etc/login.defs (debians). Takes away the permissions
5	Change Owner & Group	chown
6	Hashed passwords storage	/etc/shadow
7	Change Permissions	chmod Bit mask OGA rwx
8	Delete User	userdel, removing recusively home folder and files → userdel -r

1.3 Sudo

1. Add a user as a sudoer by using visudo. You can specify users or groups.
2. Common to have a sudo or wheel group and to give that group permissions in visudo
3. Syntax → user computerAddress=(Runas_Alias) Command_Alias

4. You can use a Runas_Alias to define a semi-super user that owns a group of files or processes. Then the user can use sudo to run as that user. Same you can limit the commands that a user can run as sudo with the Command_Alias

5. to give sudo root access use 'user' ALL=(ALL) ALL → root privilages to "user" with use of sudo

Groups

1	Wheel	Group allowing access to the sudo/su command to become another user or the superuser, for sudo this is enabled with visudo.
2	Add user to a group	usermod -a -G "group" "user"(-a only used with -G, without -a, -G makes the given groups the only additional groups he is a member of)
3	Change users primary group	usermod -g "group" "user"
4	New Group	groupadd ____
5	All groups on system	getent group
6	chgrp	change the group ownership of a file

Mounting

1	Mounting	mount /dev/____ destination
2	What disk are mounted	mount
3	Connected disks	lsblk prints out all of the connected devices nicely formatted
4	Mounting on boot	edit /etc/fstab

TAR & ZIP

1	Make a tarball	tar -cf fileout.tar filename1 filename2...
2	Extract a tarball	tar -xf filename.tar (be cautious of 'tarbombs' extract in a directory)
3	tar & gzip	tar -czf fileout.tar.gz filename1 filename2...
4	Uncompress .tar.gz	tar -xzf filename.tar.gz
5	Compress to .gz	gzip filename
6	Uncompress .gz	gzip -c filename.gz
7	Compress to .Z	compress filename
8	Uncompress .Z	uncompress filename.Z

Files

1	Types	7 types block special, char spectial, directory, normal file, symbolic link, named pipe, socket
2	diff	Get difference between 2 files or dirs
3	comm	select or reject common lines between files
4	ln	Create a symbolic link
5	link	Create a hard link

Shell Variables

1	Set a shell variable from a program output	$(arg)
2	getconf	List system config variables

Pipes & Redirection

1	Pipes	Sends the output of one file into the input of another → cat ____ — grep "____ "
2	Redirect	Use > to overwrite a file, >> to append. Use 1>> for STDOUT & 2>> for STDERR

General Bash

1	curl	Tool for talking over several different protocols

Maintenance

1	Schedule Jobs (user)	crontab, edit using crontab -e, kept in /var/spool/cron/crontabs, also package specific cron jobs are in /etc/cron.d
2	Schedule Jobs (system)	/etc/crontab
3	at	Run a process at a specified time, accepts HH:MM
4	batch	Run a process when the load drops to a specified level

1.4 Strings & Searching

Bash Strings

1	cat	Read a file
2	tac	Read a file backwards
3	nl	Number lines in output
4	Head	Read first few file lines
5	Tail	Read last few file lines
6	read	read from user input → read var → will set the var variable
7	cut	Break a line on a delimiter

1.4.1 Grep

1. Search for a character pattern in a string
2. grep ____ filename → returns the lines with the character pattern ____ in file filename
3. Follow directories "grep -r ____ ./*"
4. Get the line number → -n

5. Get files with the string → -l
6. Ignore case → -i

1.4.2 Find

1. Find a specific file by name find {Starting directory} -name "filename"
2. Finding by type → find {Starting directory} -type d/f...
3. Searching depth → find ____ -maxdepth "depth"
4. Running a command on all found files → find ____ ____ -exec "command" + (the + ends the command)
5. Files by last accessed time → -atime "days_ago" or -amin "min_ago".
   (a) a → accessed, m → modified, c → changed
   (b) use -daystart to count from the start of the current day instead of right now
   (c) use + for greater than the time, - for less and none for exactly

file

1	Find the file's character set	file -i → gives the mime type, search for
2	tac	Read a file backwards
3	Head	Read first few file lines
4	Tail	Read last few file lines
5	read	read from user input → read var → will set the var variable
6	cut -d : -f "field1"-"field2"	Break a line on a delim ':', then take the fields in range, c of chars, b bytes

2 GIT

Setup

1	Get a repo	git clone
2	Make a repo	git init
3	Pull an existing repo	Use init or clone the repo then pull
4	Remote repos	git remote → lists the remote repos, git remote add "name" "url"
5	Configuration	git config → complicated, but add email and user with git config –global user.email & user.name

3 MySQL

3.1 Users & Permissions

4 Python

5 Networking

Links: `https://github.com/kylejohnson/linux-sysadmin-interview-questions/blob/master/test.md`

6 Terms

Programming

1	Agile	See below 6

1. **Agile:** Software development strategy. Values:
   (a) **Individuals and Interactions** over processes and tools
      i. Pair programming → 1 station 2 programmers, driver & navigator/observer
      ii. Colocation → Team members in the same area
   (b) **Working Software** over comprehensive documentation
   (c) **Customer Collaboration** over contract negotiation
   (d) **Responding to Change** over following a plan