

Expansion		su / sudo (cont)		Alt commands (cont)
ls \$(which cp)		Command -ll		fdisk -l
ls `which cp`		expansion Which commands I'm allowed to execute		List partition tables
mkdir {2017...2 020 }-{ 01-12}		Brace visudo		dd bs=4M if=inp ut.iso of=/dev/vsd
echo text ~/.txt {a,b} \$USER		expansion Edit /etc/sudoers file. Example		sync
echo "text ~/.txt {a,b} \$USER"		Variable teralco ALL=(root) NOPASSWD: /etc/i		Burn iso in device
echo 'text ~/.txt {a,b} \$USER'		expansion ss		
		" teralco ALL=(j boss) NOPASSWD: /bin		Searches
		expansion (NONE)		find path -name *.log -type f -mti
		UFW		Find and delete files older than 5 days
		ufw status [verbose numbered]		find path -type f -printf "%s \t%
		Show status and rules		10
		ufw default deny incoming		Find the 10 biggest files
		ufw default allow outgoing		grep -rfile 'pattern' path
		Deny all incoming traffic by default		Find files with pattern content
		Allow all outgoing traffic by default		du -cks path/* sort -rn head -
		ufw [allow deny] from IP to [any i nte		Find biggest dirs
		ORT		du -sh path
		Full allow/deny rule		Dir size
		ufw [allow deny] servic e_name		
		Allow/deny a service (ssh www ftp ...)		Permissions
		ufw [enable disable]		chmod [u g o] [[+ - =][rwx] dir_f
		ufw delete [rule number]		ile
				Change permissions
		Alt commands		chmod u[+ -]s file
		/var/log/messages & /var/log/syslog		Set uid. File is always exec as owner user
		System log files		chmod g[+ -]s dir
		script file		Set gid. New files in folder are always
		Record session commands in file		owned by folder owner
		nohup command &		chmod [+ -]t dir
		Keep command running even after close session		Sticky Bit. Files in dir can only be renamed
		tar -Jxvf file.tar.xz [-C dest_folder]		or removed by owner or root
		Extract xz (higher compress ratio)		umask xxxx
		zip -FF x.zip --out Y.zip && unzip Y.z		Set default permissions to dirs
		Merge zip files (x.zip, x.z01, x.z02)		
		ps aux --sort -rss		
		Higher memory consumption processes		

Enviroment

<code>/etc/profile</code>	Global env config files
<code>/etc/bash.bashrc</code>	
<code>./bashrc</code>	User env config file
<code>./profile</code>	
<code>printenv</code>	Print defined env vars
<code>alias</code>	Show defined aliases
<code>export VAR</code>	Make a VAR available to child process
<code>source script</code> <code>. script</code>	Load a file into current script or shell session

Network

<code>ip addr</code>	Show ips
<code>ifup ifdown interface_name</code>	Up or down an interface
<code>/etc/network/interfaces</code>	File interfaces are defined
<code>hostnamectl [set-hostname hostname]</code>	Manage hostname (/etc/hostname) without restarting
<code>netstat -tuln</code>	All tcp and udp listening ports
<code>ss -tuln</code>	All tcp and udp listening ports
<code>nc -vz host port</code>	Scan port in host

SSH

<code>ssh-keygen [-f /etc/ssh/ssh_host_rsa_key] -t rsa -b 4096</code>	Generate ssh rsa key. By default <code>~/.ssh/id_rsa</code>
<code>ssh-copy-id [-i ~/.ssh/mykey] user@host</code>	Copy ssh key in remote host (<code>~/.ssh/authorized_keys</code>). By default <code>~/.ssh/id_rsa</code> .
<code>ssh user@host command</code>	Exec command in host
<code>ssh -J hostA hostB</code>	Connect to hostB through hostA (Jumping). Identification in localhost. Can use ProxyJump in <code>~/.ssh/config</code>
<code>ssh -D port -fCqN user@host</code>	Proxy socket (HTTP and HTTPS traffic) through port.
<code>ssh -nNT -L 9000:remote_server:80 user@host</code>	Tunneling (Local port forwarding). Map <code>remote_server:80</code> into <code>localhost:9000</code> through host
<code>ssh -nNT -R 9000:localhost:3000 user@host</code>	Tunneling (Remote port forwarding). Map <code>localhost:3000</code> into <code>host:9000</code>