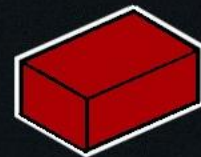


HANDS UP IF YOU DON'T HAVE A  
VM  
OR IF YOU DON'T REMEMBER  
YOUR PASSWORDS

Or something broke



**Redbrick**  
DCU's Networking Society

# Quick recap from last week.

- Fixed all the network issues (we think)

# Access Over VNC

- Open a terminal
- Use the ssh command.
- Now on your local pc,
- Open VNC client and connect to localhost port 5900

**Redbrick VM System** [Home](#) | [Logged in as testing](#) | [Preferences](#) | [Logout](#)

## VM Power On

VM power on successful

**To connect to your virtual machine's VNC server:**

1. Connect to redbrick, SSH forwarding port 5900 to 136.206.16.1:5913:  

```
ssh -L 5900:136.206.16.1:5913 username@login.redbrick.dcu.ie
```
2. Open your VNC client, and connect to localhost display 0 (or port 5900).
3. Enter the following password when prompted: **Ym1esolk**
4. **Keep this password in a safe place. It will be valid while your VM is running, and it will not be displayed to you again after this point. To reset it, you'll have to power off and power on your virtual machine from this web interface.**

Click [here](#) to return to the VM management screen.

Redbrick VM Management System - Contact [admins@redbrick.dcu.ie](mailto:admins@redbrick.dcu.ie) for help or bug reports - Source available at <http://hg.redbrick.dcu.ie/rbvm>.

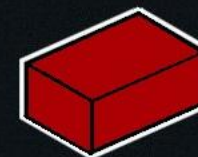


# Access over SSH

- Log in to redbrick via command line
- ssh to your vm using your username (on the vm) and your vm ip address (gotten from <https://vmweb.redbrick.dcu.ie/login>)

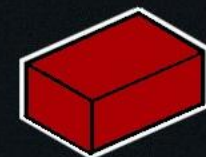
## Now that the network works...

- Login into your redbrick account and ssh to your vm using the login details you created.
- “ssh \$your\_username@\$your\_vm\_IP”
  
- Because of your firewalls you can only ssh to your vm from redbrick.
- Now that we have access over ssh
- Lets update the package lists to the newest versions
  
- run “apt-get update”



## Su > Sudo:

- **Sudo** : allows any user in the sudoers file to run commands as root provided they have the sudoers passwd.
  - elevates your permissions in your current shell, (kind of blurs the lines between the “user” and “root”)
- **Su**: Allows users to login as root. They have full control of the operating system. Maintains root until the user logs out of root. (Or they timeout)





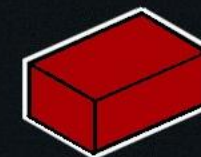
# Intruder Alert: Fending off the bot-nets

- First we need a new root password.
    - to change your password type “passwd” Don't forget this password
  - Now that we have a secure root password lets disable root ssh and users ability to sudo(including you).
- 1) First we need to find the config files
  - 2) Next we need to edit the config file to disable root
  - 3) Then we need to find the sudoers file.

Anyone any ideas?

## Intruder Alert: Fending off the bot-nets

- So the first file we need is the `sshd_config` which is found in `/etc/ssh`.
- Once we have found the file we need to change `"PermitRootLogin yes"` to `"PermitRootLogin no"`
- The next file we need is `/etc/sudoers`
- By Default `/etc/sudoers` is read only and can only be edited by using the command `"visudo"`
- we want to comment out the line that says  
`"%sudo ALL=(ALL:ALL) ALL"`
- To comment out the line put a `#` in front of it and save the file





# HTOP > TOP :

```
top - 13:52:27 up 36 days, 22:42, 21 users, load average: 3.22, 2.67, 2.63
Tasks: 437 total, 5 running, 424 sleeping, 7 stopped, 1 zombie
Cpu(s): 38.2%us, 0.7%sy, 0.0%ni, 61.0%id, 0.0%wa, 0.0%hi, 0.1%si, 0.0%st
Mem: 8192616k total, 7430948k used, 761668k free, 393348k buffers
Swap: 11408376k total, 1161380k used, 10246996k free, 3999588k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
15920	snow	20	0	421m	174m	33m	R	100	2.2	0:16.38	php-cgi
15954	snow	20	0	419m	172m	33m	R	100	2.2	0:12.41	php-cgi
4870	list	20	0	534m	497m	3120	R	100	6.2	2833:12	python
15978	snow	20	0	314m	67m	32m	R	13	0.8	0:00.41	php-cgi
16123	edu	20	0	11752	1668	1160	S	3	0.0	120:03.36	htop
15976	awhshit	20	0	11388	1548	956	R	1	0.0	0:00.10	top
25927	fun	20	0	99.2m	35m	3128	S	1	0.4	45:13.40	irssi
4876	list	20	0	64944	25m	2952	S	0	0.3	6:54.42	python
19827	awhshit	20	0	91592	876	752	S	0	0.0	0:01.54	sshd
1	root	20	0	61900	2340	1720	S	0	0.0	1:04.18	init
2	root	20	0	0	0	0	S	0	0.0	0:00.20	kthreadd
3	root	RT	0	0	0	0	S	0	0.0	0:16.54	migration/0
4	root	20	0	0	0	0	S	0	0.0	4:13.11	ksoftirqd/0
5	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/0
6	root	RT	0	0	0	0	S	0	0.0	0:32.95	migration/1
7	root	20	0	0	0	0	S	0	0.0	2:52.44	ksoftirqd/1
8	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/1
9	root	RT	0	0	0	0	S	0	0.0	0:21.40	migration/2
10	root	20	0	0	0	0	S	0	0.0	0:45.91	ksoftirqd/2
11	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/2
12	root	RT	0	0	0	0	S	0	0.0	0:06.63	migration/3
13	root	20	0	0	0	0	S	0	0.0	0:26.25	ksoftirqd/3
14	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/3
15	root	RT	0	0	0	0	S	0	0.0	0:16.00	migration/4
16	root	20	0	0	0	0	S	0	0.0	1:34.00	ksoftirqd/4
17	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/4
18	root	RT	0	0	0	0	S	0	0.0	1:01.77	migration/5
19	root	20	0	0	0	0	S	0	0.0	1:23.34	ksoftirqd/5
20	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/5
21	root	RT	0	0	0	0	S	0	0.0	0:28.15	migration/6
22	root	20	0	0	0	0	S	0	0.0	0:46.47	ksoftirqd/6
23	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/6
24	root	RT	0	0	0	0	S	0	0.0	0:08.58	migration/7

```
[ azazel ] [ 0- Chat 1 zsh 2! Azazel 3! B4 4 Daniel 5 Thunder 6 pyg (7*zsh)
```

```
][ 18/11/14 13:52 ]
```

# This is HTOP

```

1  [||||||||||||||||||||||||||||||||||||||||||||||||||||| 89.5%]
2  [||||||||||||||||||||||||||||||||||||||||||||||||| 73.4%]
3  [||||||||||||||||||||||||||||||||||||||||||||||||| 95.5%]
4  [||||||||||||||||||||||||||||||||||||||||||||| 36.1%]
5  [||||||||||||||||||||||||||||||||||||||||||||||||| 96.7%]
6  [||||||||||||||||||||||||||||||||||||||||||||||||| 100.0%]
7  [||| 1.3%]
8  [||| 3.1%]
Mem[||||||||||||||||||||||||||||||||||||||||| 3155/8000MB]
Swp[||||| 1134/11140MB]

Tasks: 374 total, 7 running
Load average: 3.14 2.69 2.64
Uptime: 36 days, 22:43:05

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
16065 snow      20   0  414M  167M 34124 R 100.  2.1  0:03.66 /usr/bin/php-cgi
4870 list      20   0  534M  497M  3120 R 97.0  6.2 47h13:31 /usr/bin/python /var/lib/mailman/bin/qrunner --runner=ArchRunner:0:1 -s
16067 snow      20   0  412M  165M 34124 R 97.0  2.1  0:01.94 /usr/bin/php-cgi
16069 snow      20   0  406M  159M 34088 R 62.0  2.0  0:01.40 /usr/bin/php-cgi
9462 mysql    20   0  427M  100M  5104 S  5.0  1.3 15:06.37 /usr/sbin/mysqld
11508 mysql    20   0  427M  100M  5104 S  5.0  1.3  1:20.71 /usr/sbin/mysqld
17637 mysql    20   0  427M  100M  5104 S  4.0  1.3  6:21.93 /usr/sbin/mysqld
16064 awshsh   20   0 11924  1544  1004 R  3.0  0.0  0:00.31 htop
16123 edu      20   0 11752  1668  1160 S  3.0  0.0  2h00:03 htop
11506 mysql    20   0  427M  100M  5104 S  0.0  1.3  1:32.59 /usr/sbin/mysqld
8553 mysql    20   0  427M  100M  5104 S  0.0  1.3 12:33.00 /usr/sbin/mysqld
  1 root     20   0 61900  2340  1720 S  0.0  0.0  1:04.18 /sbin/init
 429 shadow  39  19  4120   548   488 S  0.0  0.0  8:25.54 /bin/sh /home/associat/s/shadow/caridle
 505 shadow  24   4  4120   556   508 S  0.0  0.0  1:08.60 /bin/sh /home/associat/s/shadow/activate
 533 root     20   0 17060   568   468 S  0.0  0.0  0:00.07 upstart-udev-bridge --daemon
 558 root     16  -4 17120   240   236 S  0.0  0.0  0:00.05 udevd --daemon
 592 butlerx 20   0  4120   484   480 T  0.0  0.0  0:00.01 sh -c nano /tmp/sqlrIoIUT
 593 sonic    20   0 67324  4692   964 S  0.0  0.1  7:17.34 SCREEN -DRR
 594 sonic    20   0 22880  1320   960 S  0.0  0.0  0:00.24 -/usr/local/shells/zsh
 595 sonic    20   0 22296   776   772 S  0.0  0.0  0:00.04 -/usr/local/shells/zsh
 596 sonic    20   0 22296   776   772 S  0.0  0.0  0:00.04 -/usr/local/shells/zsh
 598 sonic    20   0 26760  1932  1388 S  0.0  0.0  0:00.89 -/usr/local/shells/zsh
 599 butlerx 20   0 11396   780   776 T  0.0  0.0  0:00.04 nano /tmp/sqlrIoIUT
 886 daemon   20   0  8280   552   456 S  0.0  0.0  0:00.15 portmap
1385 statd   20   0 16696   764   760 S  0.0  0.0  0:00.02 rpc.statd -L
1445 messageb 20   0 59820  1604   984 S  0.0  0.0  0:02.93 dbus-daemon --system --fork

F1 Help F2 Setup F3 Search F4 Invert F5 Tree F6 SortBy F7 Nice - F8 Nice + F9 Kill F10 Quit
[ azazel ][ 0- Chat 1 zsh 2! Azazel 3! B4 4 Daniel 5 Thunder 6 pyg (7*zsh) ] [ 18/11/14 13:52 ]

```



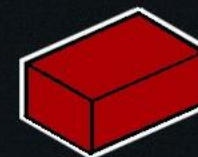
## Let's install HTOP:

Run: "apt-get install htop"

Some useful htop flags

-u : allows you to view the output for a particular user. - "htop -u \$username"

-d : allows you to delay the time between updates (10ths of seconds) "htop -d 15"



**Redbrick**  
DCU's Networking Society



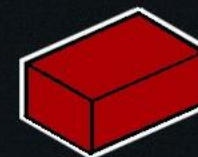
## Old friends GREP and | (PIPE)

Grep: Allows you to search for patterns in a commands output.

| : Allows you to string commands together.

“cat /etc/network/interfaces” - long output

“cat /etc/network/interfaces | grep gateway” - only the output we want

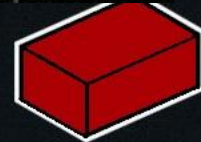


**Redbrick**  
DCU's Networking Society

**YO DAWG, I HEARD YOU LIKE TO |GREP**

**SO I PUT A |GREP IN YOUR |GREP SO  
YOU CAN |GREP WHILE YOU |GREP**

quickmeme.com



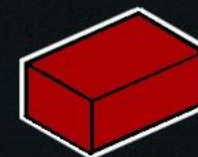
**Redbrick**  
DCU's Networking Society



## But what if I'm lazy? .....ALIASES

Aliases allow you you to be very very lazy.

- You can write synonyms for commands.
- "SshBrick" instead "ssh \$[username@login.redbrick.dcu.ie](https://login.redbrick.dcu.ie):"



**Redbrick**  
DCU's Networking Society



## Lets write some aliases

### I want to get back to my home directory quickly

home - "cd /home/username"

type > alias home='cd /home/username'

### I'm wondering what processes I am running at the moment.

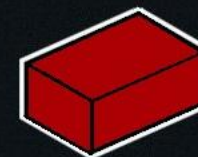
whatdo - "ps aux | grep \$username"

type > alias whatdo='ps aux | grep kylar'

### I want to ssh to redbrick from my VM

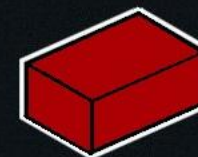
sshbrick "ssh \$username@login.redbrick.dcu.ie"

type > alias sshbrick='ssh \$username@login.redbrick.dcu.ie'



**Redbrick**  
DCU's Networking Society

# Cron



**Redbrick**  
DCU's Networking Society

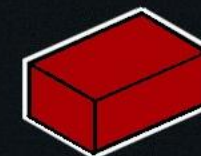
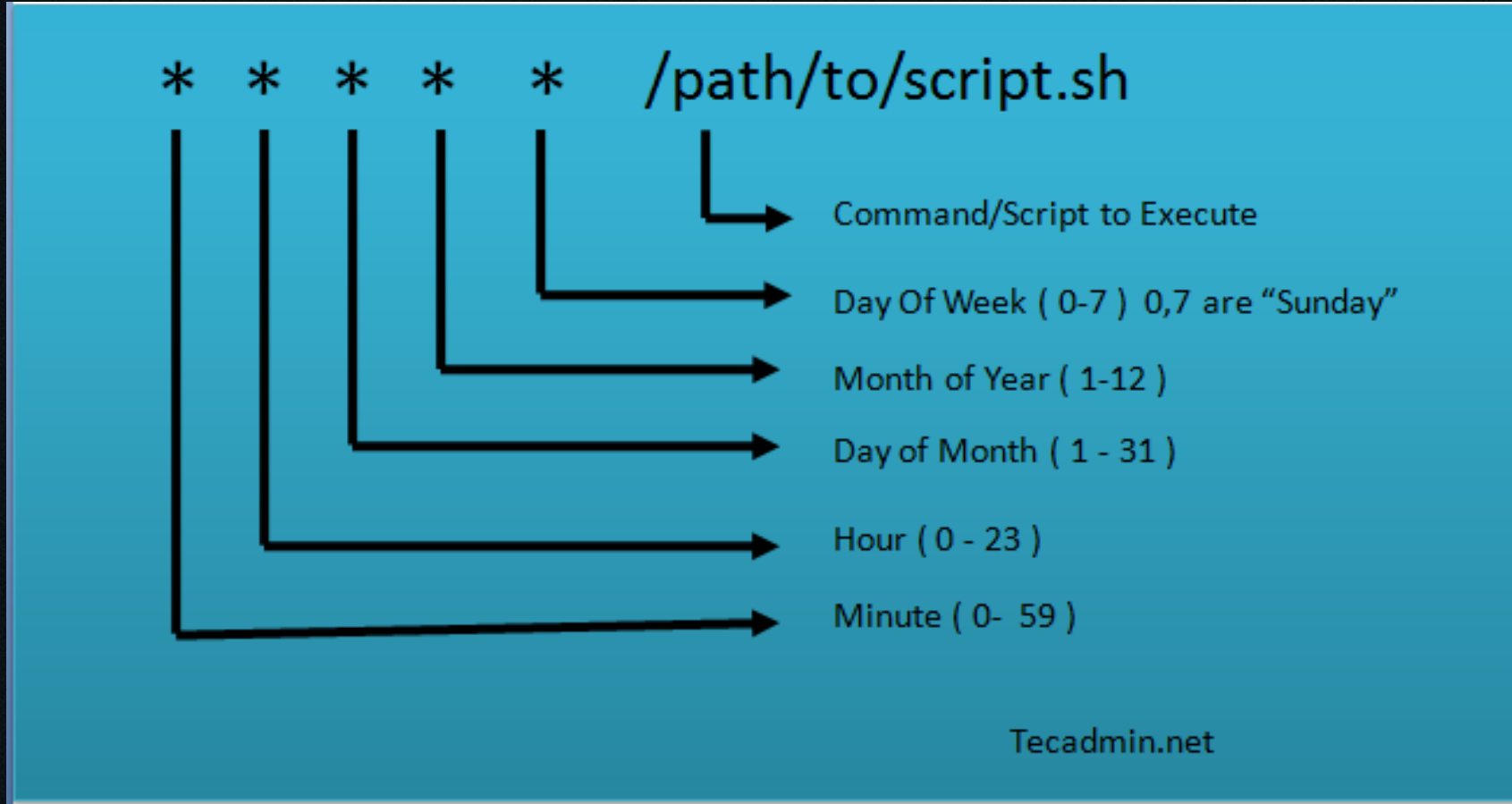


# Cron

- The cron daemon allows you to run commands/scripts at regular intervals.
  - On certain days, hours, months, weeks.
- So what would I use it for?
  - Running backups
  - Copying files to remote servers
  - Sending automated emails
- Run “crontab -e”

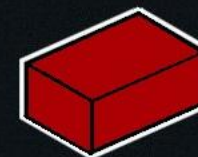


# Cron Syntax



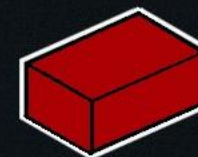
## Writing your first cron job.

- We're going to backup our home directories every day.
- Type `crontab -e`
  
- Read the instructions,
- Your cron line should look like
  - `0 0 * * 0 tar -zcf /var/backups/home.tgz /home/`
  - This will back-up all user accounts at 12am every Sunday morning





**KEEP  
CALM  
AND  
ASK ME  
QUESTIONS**



**Redbrick**  
DCU's Networking Society



# Tell us what you want?

- Go through sections of the admin exam?
  - Security
  - Filesystems
  - Networking
  - Hardware
  - Practical/Scripting
- Do more things with vms?
  -
- Something completely different?
- More in depth command like talk?

