‹ (../02-logging-onto-cloud/index.html) Introduction to Cloud Computing for Genomics (../) › (../04-which-cloud

Fine tuning your Cloud Setup

Overview

Teaching: 5 min
Exercises: 10 min
Questions

- Is my remote computer correctly configured?
- How do I keep my processing going when I leave?

Objectives

- Check the available resources and file system on your remote machine
- Keep background processes working in the cloud with `tmux`

Is this the right cloud?

Once you're connected to your new remote instance, it's a good idea to double check that the settings are what you wanted, and that everything is working smoothly before you start your project.

For this workshop, your instructor did all the verification before the workshop even started, but this is an important skill for when you start running your own instances.

Verifying your connection

When you connect, it is typical to receive a welcome screen. The Data Carpentry Amazon instances display this message upon connecting:

Output

```
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-48-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Tue Jan 29 22:28:18 UTC 2019

  System load:  0.04               Processes:           164
  Usage of /:   43.1% of 98.30GB   Users logged in:     0
  Memory usage: 2%                 IP address for eth0: 172.31.41.107
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

You should also have a blinking cursor awaiting your command

Bash

```
$
```

Verifying your environment

Now that we have connected here are a few commands that tell you a little about the machine you have connected to:

- `whoami` - shows your username on computer you have connected to:

  Output

  ```
  dcuser@ip-172-31-62-209 ~ $ whoami
  dcuser
  ```

- `df -h` - shows space on hard drive

  Output

  ```
  dcuser@ip-172-31-62-209 ~ $ df -h
  Filesystem      Size  Used Avail Use% Mounted on
  udev            2.0G   12K  2.0G   1% /dev
  tmpfs           396M  792K  395M   1% /run
  /dev/xvda1       99G   48G   47G  51% /
  none            4.0K     0  4.0K   0% /sys/fs/cgroup
  none            5.0M     0  5.0M   0% /run/lock
  none            2.0G  144K  2.0G   1% /run/shm
  none            100M   36K  100M   1% /run/user
  ```

  Under the column 'Mounted on row' that has `/` as the value shows the value for the main disk

- `cat /proc/cpuinfo` - shows detail information on how many processors (CPUs) the machine has

  Output

```
dcuser@ip-172-31-62-209 ~ $ cat /proc/cpuinfo
processor  : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 62
model name      : Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz
stepping        : 4
microcode       : 0x415
cpu MHz         : 2494.060
cache size      : 25600 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse
sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology eagerfpu pni pclmulqdq ssse3 cx16 pcid sse4_1
sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm xsaveopt fsgsbase smep erm
s
bogomips        : 4988.12
clflush size    : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
power management:

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 6
model           : 62
model name      : Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz
stepping        : 4
microcode       : 0x415
cpu MHz         : 2494.060
cache size      : 25600 KB
physical id     : 0
siblings        : 2
core id         : 1
cpu cores       : 2
apicid          : 2
initial apicid  : 2
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse
sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology eagerfpu pni pclmulqdq ssse3 cx16 pcid sse4_1
sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm xsaveopt fsgsbase smep erm
s
bogomips        : 4988.12
clflush size    : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
power management:
```

- `tree -L 1` - shows a tree view of the file system 1 level below your current location.

Output

```
dcuser@ip-172-31-62-209 ~ $ tree -L 1
.
├── dc_sample_data
├── Desktop
├── Downloads
├── FastQC
├── openrefine-2.6-beta.1
├── R
└── Trimmomatic-0.32

7 directories, 0 files
```

Staying Connected to the Cloud

Depending on how you connect to the cloud, you may have processes and jobs that are running, and will need to continue running for some time. If you are connecting to your cloud desktop via VNC, jobs you start will continue to run. If you are connecting via SSH, if you end the SSH connection (e.g. you exit your SSH session, you lose your connection to the internet, you close your laptop, etc.), jobs that are still running when you disconnect will be killed. There are a few ways to keep cloud processes running in the background. Many times when we refer to a background process we are talking about what is described at this tutorial (http://www.cyberciti.biz/faq/linux-command-line-run-in-background/) - running a command and returning to shell prompt. Here we describe a program that will allow us to run our entire shell and keep that process running even if we disconnect: `tmux`. If you don't have `tmux` on your system, you should still be able to use `screen`. This is another program that has mostly the same capabilities as `tmux`. It's a lot older, though, so can be more clunky to use; however, it is likely to be available on any cloud system you encounter.

In both `tmux` and `screen`, you open a 'session'. A 'session' can be thought of as a window for `tmux` or `screen`, you might open an terminal to do one thing on the a computer and then open a new terminal to work on another task at the command line.

As you work, an open session will stay active until you close this session. Even if you disconnect from your machine, the jobs you start in this session will run till completion.

For the following instructions use either `tmux` OR `screen`, not both!

Starting and attaching to a session

You can start a session and give it a descriptive name:

- `tmux`

  Bash

  ```
  $ tmux new -s session_name
  ```

- `screen`

  Bash

  ```
  $ screen -S session_name
  ```

This creates a session with the name `session_name` which will stay active until you close it.

Detach session (process keeps running in background)

You can detach from a session by pressing on your keyboard:

- `tmux`: `control + b` followed by `d` (for detach)
- `screen`: `control + a` followed by `d` (for detach)

Seeing active sessions

If you disconnect from your session, or from your ssh into a machine, you will need to reconnect to an existing session. You can see a list of existing sessions:

- `tmux`

```
Bash

$ tmux list-sessions
```

- `screen`

```
Bash

$ screen -ls
```

Connecting to a session

To connect to an existing session:

- `tmux`

```
Bash

$ tmux attach -t session_name
```

The `-t` option = 'target'

- `screen`

```
Bash

$ screen -r session_name
```

The `-r` option = 'resume a detached screen session'

Switch sessions

You can switch between sessions:

- `tmux`

```
Bash

$ tmux switch -t session_name
```

Kill a session

You can end sessions:

- `tmux`

```
Bash

$ tmux kill-session -t session_name
```

- `screen`

Bash

```
$ screen -r session_name
$ exit
```

Installing additional software

By default `tmux` is not installed in most cloud Linux instances. However when you start a new instance, you can install new software packages using Package Managers like YUM (for Red Hat and Centos instances) or APT (for Debian or Ubuntu instances).

📌 Caution

In this lesson, you are using an Amazon instance owned by someone else, so you *won't* be able to install packages, but we'll show you how to use APT (Advanced Package Tool) to find packages, so you know how to do it when you launch your own instance.

Search for APT packages using including software

Most common software tools will have a package named with the same name, but this is not always the case. If you know the name of the program you wish to install, but are not sure of the package name, you can use the `apt` program to search packages:

Bash

```
$ apt search tmux
Sorting... Done
Full Text Search... Done
tmux/trusty,now 1.8-5 amd64 [installed]
  terminal multiplexer
$
```

On our system, searching for tmux only gives one result, and it is already installed.

🖉 Exercise

Check to see whether APT can be used to install your favorite bioinformatics program, or ones you commonly see used in your field. If you can't think of anything, try to search for BLAST. What do you have to search for to get back the results you'd expect?

Install packages using APT

If you own, or at least have administrator privileges on an instance, you can also use APT to install a package. First you would need the package name, which is whatever is before the `/` in your search result. In our example above with `tmux`, we got

Bash

```
tmux/trusty,now 1.8-5 amd64 [installed]
```

which means that it is stored in APT as tmux.

Exercise

What are the package names for programs you need in your pipeline? Are they the same as the program name? What package name would you need to install the old version of BLAST?

Once you know the package name, you could install it using `apt install`:

Reminder

Remember, the following instructions are for demonstration only, you don't have the administrator password needed to run these commands on your workshop instance.

Update APT

Before installing or upgrading any system packages, you should always update the local APT cache. That ensures you'll install the latest version. Note that the instructions now start with `sudo`, which is short for 'super user do'. `sudo` is a program that allows a user to users to run programs as an administrator without logging off and then logging back in as the admin.

So, in this line:

Bash

```
$ sudo apt upgrade
```

we are first invoking `sudo`, and then having the `sudo` program run `apt upgrade`. This way, `apt upgrade` is run from the administrator account. You can actually try to run that line if you want, you'll be prompted to input the administrator password:

Bash

```
$ sudo apt upgrade
[sudo] password for dcuser:
```

Since we don't have the adminstrator password, our request will be rejected:

Bash

```
dcuser@ip-172-31-26-134:~$ sudo apt upgrade
password for dcuser:
dcuser is not in the sudoers file.  This incident will be reported.
dcuser@ip-172-31-26-134:~$
```

If we *did* have the administrator password, we would have seen this:

Bash

```
$ sudo apt upgrade
password for dcuser:
Hit:1 http://au.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://au.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
...
Fetched 3,413 kB in 1s (2,233 kB/s)
Reading package lists... Done
```

And one the cache is updated, we could have then requested that APT install a program. To have APT find packages, we used `apt search`, which told the program APT to run it's sub-program 'search' but to install packages, we need to use the subprogram 'install'. Confusingly, there is also an `apt-get` program with an 'install' subprogram which does exactly the same thing, in 99% of cases, it doesn't matter whether you use `apt install` or `apt-get install`.

As with the 'upgrade' command, this will require the administrator password that we don't have, but on your own machine, you'd get output like this:

Bash

```
$ sudo apt install tmux
password for dcuser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libevent-2.0-5 libutempter0
The following NEW packages will be installed:
  libevent-2.0-5 libutempter0 tmux
0 to upgrade, 3 to newly install, 0 to remove and 0 not to upgrade.
Need to get 345 kB of archives.
After this operation, 949 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirror.overthewire.com.au/ubuntu xenial-updates/main amd64 libevent-2.0-5 amd64 2.0.21-stable-2ubuntu
0.16.04.1 [114 kB]
Get:2 http://mirror.overthewire.com.au/ubuntu xenial/main amd64 libutempter0 amd64 1.1.6-3 [7,898 B]
Get:3 http://mirror.overthewire.com.au/ubuntu xenial/main amd64 tmux amd64 2.1-3build1 [223 kB]
Fetched 345 kB in 0s (863 kB/s)
(Reading database ... 130583 files and directories currently installed.)
...
Setting up libevent-2.0-5:amd64 (2.0.21-stable-2ubuntu0.16.04.1) ...
Setting up libutempter0:amd64 (1.1.6-3) ...
Setting up tmux (2.1-3build1) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
```

Key Points

- Always check a new instance to verify it started correctly
- Using a program like `tmux` can keep your work going even if your internet connection is bad

‹ (../02-logging-onto-cloud/index.html)

› (../04-which-cloud