Getting started with CloudBioLinux

Bela Tiwari

August 16, 2011

Contents

Chapter 1

Preparing to work on the Amazon Cloud

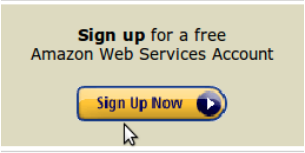


Figure 1.1: Signing up for an AWS account starts with the click of a button.

1.1 Get an Amazon AWS account

Anyone can set up an account with Amazon to access their computer cloud. Just go to http://aws.amazon.com and sign up for an account.

The rest of this document assumes you have an AWS account and you are logged into it.

1.2 Get an Amazon EC2 Account

There are various ways you can access the power of the Amazon cloud. In this document, we describe using EC2.

If you do not already have one, you need to sign up for an EC2 account. This is in addition to the general Amazon aws account you have if you followed the instructions in section 1.1.

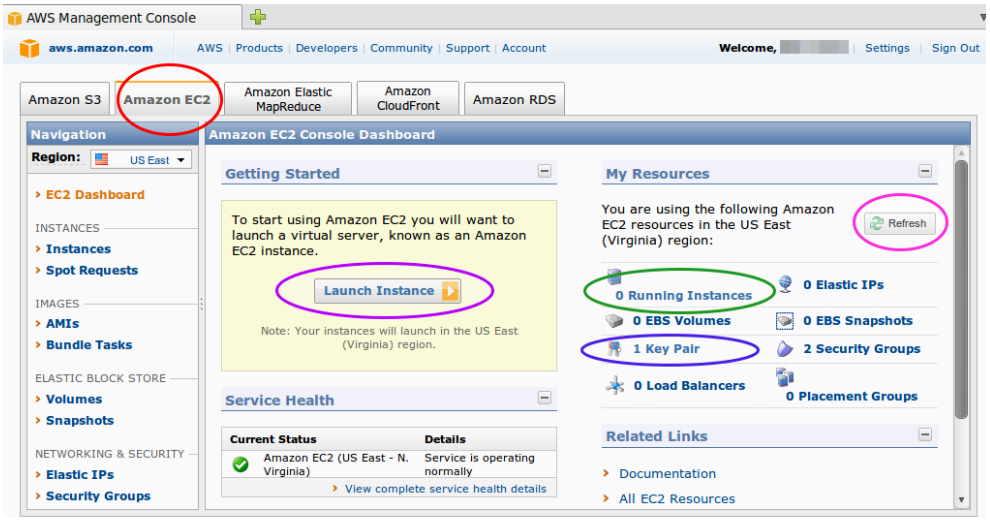
To get your EC2 account ,

1. click on the Products tab on the Amazon aws page,
2. click on the Amazon Elastic Compute Cloud (EC2) link that appears in the Compute section of the listing, and
3. click on the button in the right hand pane that says Sign up for Amazon EC2.
4. Complete the registration process[1].

1.3 Get an EC2 key pair

After you've signed up for your account, Amazon will send you an email with a link in it to the Access Identifiers section of your account. Amazon provides a list of which credentials you need to do particular tasks. If all you will be doing is starting up Bio-Linux using the Amazon (graphical) console, then you only need an Amazon EC2 Key Pair.

[1] Signing up for Amazon EC2 also automatically signs you up for Amazon Simple Storage Service and Amazon Virtual Private Cloud. You will not be charged for any service unless you use it.



To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (Virginia) region.

Current Status	Details
Amazon EC2 (US East - N. Virginia)	Service is operating normally

View complete service health details

You are using the following Amazon EC2 resources in the US East (Virginia) region:

Figure 1.2: Your AWS home are will look something like this. Here, we are looking at the information under the EC2 tab (Red Circle). Blue Circle: your Key Pairs - if you don't have any, click on the link to create some. Purple Circle: You can easily launch any publicly available EC2 image from Amazon by clicking on this button. Green Circle: The number of running instances you have. If you have any, you can find out more about them by clicking on this link. Pink Circle: If you add Key Pairs, or start up instances, you may need to hit the refresh button to see changes on your EC2 Dashboard.

To create a key pair:

1. Go to the the EC2 area on Amazon https://console.aws.amazon.com/ec2/home.
2. Click on the *Key Pairs* link under My Resources in the right hand area of the window. See the blue circle in figure 1.2.
3. Click on the *Create Key Pair* button near the top of the Key Pairs section of the window. See figure 1.3 on page 5.
4. Give your key pair a memorable name when prompted. Save your private key to a safe location. See the further information below about this.
5. Click on the link in the left hand pane to go back to the *EC2 Dashboard* and then click on the *Refresh button* at the far right hand side of the window (see pink circle in figure 1.2).

You should now see that you have a key pair registered in the *My Resources* section.

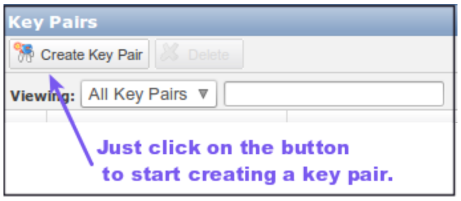


Figure 1.3: Keypair creation is simple - just click on the button and follow the instructions on screen.

Each EC2 key pair includes a private key file and a public key file. **Save your private key to a secure and memorable location**. Don't lose it or share it. (Amazon does not make a copy of it.) If you're working on Linux, adjust the permissions on your key file so it is readable only by you.

If you plan to use the command line tools to start up an instance, you will also need to get your X509 certificates. This document assumes that you will only be using the graphical console, so this is not covered further here.

How the EC2 key pairs work

When you launch a CloudBioLinux instance from Amazon, you will specify a particular EC2 key pair. The Amazon system puts a copy of your public key, which it has a record of, on the instance. You, as the (only!) holder of the private key will be the only one able to access the CloudBioLinux instance you just started up.

1.4 Install an NX Client on your local machine

To run a full desktop session, you need have NX client software on the machine you will be connecting to the Amazon Cloud from. If you don't already have NX Client software installed, we recommend the free clients from NoMachine[2].

You only need to install an NX Client on your local system if you wish to run a full graphical desktop session for working on CloudBioLinux.

[2] CloudBioLinux already has the FreeNXServer software installed

Chapter 2

Running a CloudBioLinux instance

2.1 The process in a nutshell

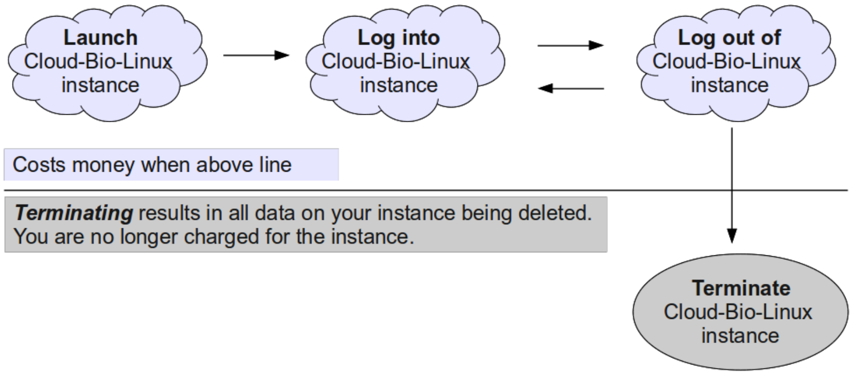

Launch Cloud-Bio-Linux instance
Log into Cloud-Bio-Linux instance
Log out of Cloud-Bio-Linux instance
Costs money when above line
Terminating results in all data on your instance being deleted. You are no longer charged for the instance.
Terminate Cloud-Bio-Linux instance

Figure 2.1: Schematic of the basic process of working on CloudBioLinux on Amazon EC2. You can log in and out of your instance as often as you like; you continue to be charged whether you are logged in or not. You can also stop and re-start instances (not shown here). When you are finished with an instance, you should terminate it. Once terminated, the system and all data on it are deleted. This is the simplest setup. There are additional easy steps you can take to store data and even whole systems, at a fraction of the price of a running image.

The general process you will follow when working with CloudBioLinux is outlined in figure 2.1:

1. **Start up** a CloudBioLinux instance

2. **Log into** your CloudBioLinux instance

3. **Log out of** the CloudBioLinux instance

4. Still want to work on this instance? You can log into it and out of it as often as you like, or you can stop and start the instance, which can work out slightly cheaper.

5. When you're really finished, and don't need the CloudBioLinux instance anymore, **terminate** the instance. You will stop being charged for this instance when it is terminated.[1]

[1] Stopping and terminating are different.

This chapter focusses on starting and logging into a full graphical Bio-Linux desktop on the cloud. Of course, there are other things that you may wish to do, like save your system and data for use again later, or share it with others. These things and more are covered in the http://aws.amazon.com/ebs/official documentation for Elastic Block Storage.

A note about charging:

The charging structure for Amazon EC2 is well defined and quite detailed. It is important to understand what you are being charged for, so you can make good decisions about when using the cloud is a cost effective option, and when it is not. You will be charged for running instances, and also for things like bandwidth when transferring data on and off Amazon systems, and data volumes you wish to use later. Please read the Amazon pricing documentation so you don't get surprised when you next see your credit card bill.

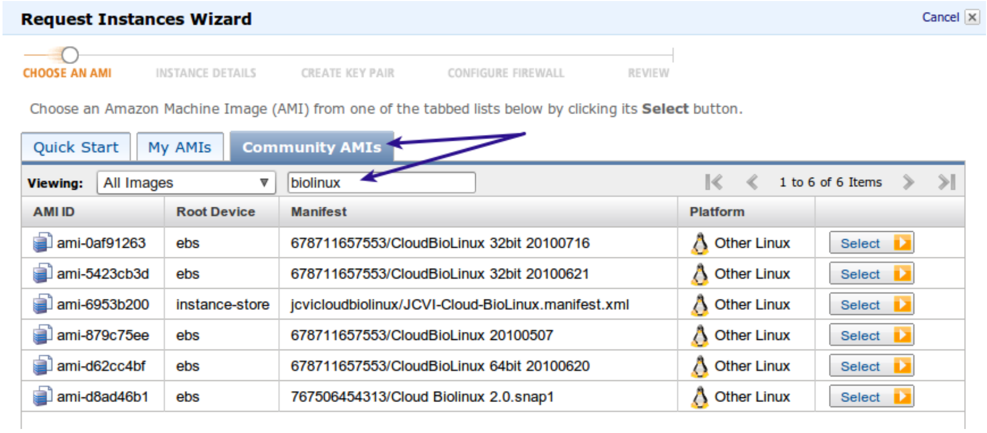
A couple of things to note when starting out:

- **You will be charged for the time your instance is running.** It's not about when you're logged into it that counts. Charging for the instance terminates when you terminate the instance. You can just transfer your files off the system onto your local machine - but be aware that you may be charged for the bandwidth you use. Alternatively, you could consider using Elastic Block Storage.
- **You are charged by the time-hour.** This means that if you start up an instance at 1:55pm and use it until 2:05pm, you are charged for two hours - because your instance was running in two different hours of the clock.

2.2 Starting up a CloudBioLinux instance

This document focusses on using the AWS Management Console, a web-interface, for managing Amazon Web Services resources.

1. Go to the EC2 Management Console URL: http://console.aws.amazon.com/ec2/home
2. You should see a button saying **Launch instance**. Click on this.
3. You are presented with a window called **Request Instances Wizard**.
4. To start up CloudBioLinux, go to the **Community AMIs tab** and search All Images for the term **biolinux**. This will bring up a list of available CloudBioLinux images. See figure 2.2.
5. Click on the image you wish to run to highlight it. Then click on the Select button on the right hand side.



Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Viewing: All Images ▼ biolinux

AMI ID	Root Device	Manifest	Platform	
ami-0af91263	ebs	678711657553/CloudBioLinux 32bit 20100716	Other Linux	Select
ami-5423cb3d	ebs	678711657553/CloudBioLinux 32bit 20100621	Other Linux	Select
ami-6953b200	instance-store	jcvicloudbiolinux/JCVI-Cloud-BioLinux.manifest.xml	Other Linux	Select
ami-879c75ee	ebs	678711657553/CloudBioLinux 20100507	Other Linux	Select
ami-d62cc4bf	ebs	678711657553/CloudBioLinux 64bit 20100620	Other Linux	Select
ami-d8ad46b1	ebs	767506454313/Cloud Biolinux 2.0.snap1	Other Linux	Select

Figure 2.2: Search for the term ”biolinux” in the Community instances. You are likely to find a number of different images available. Those listed here were available on July 20, 2010. We recommend you pick the one with the system you want (e.g. 32 bit or 64 bit) that has **the most recent date in the text within the Manifest column**.

6. Now click on the **Launch Instances** in the next window presented to you.

7. Click on the Continue Button at the bottom of the next window.

8. Leave the Advanced options on the next page as they are. *Note that this would be the time to alter these settings if you wanted to do so; you cannot change them in a running instance.*

9. In the next window, you'll need to provide the name of your Key Pair. *If you created a key pair earlier, but are not offered the option of using it, and if you created your keys in the same session you are currently logged into, try logging out of Amazon and logging back in again.*

10. Once you have provided a key pair name, the next window will ask about your preferred security settings. This is analagous to setting up a machine firewall. **At a minimum you will need to enable ssh access;** ssh is the protocol you need to use to connect to your instance, whether you do so via the command line or via a graphical NX connection. If you want to access web pages provided by your instance, then you also need to open a port for http. You will want to do this if, say, you wish to refer to the Bio-Linux documentation pages on your instance. If you will be running MySQL or postgreSQL for example, you'll need to enable access to these also.

11. Once you've done all this, you should be able to review the information

you've provided, and if you're happy click on the **Launch instance** button.

If you go back to your Amazon EC2 home area and click on the Instances link in the left hand pane, you should see your CloudBioLinux instance starting up. When you see a green icon with the word running beside it, your instance is ready to log into.

2.3 Connecting to and logging into your CloudBioLinux instance

2.3.1 Graphical, or command line?

For those wanting to work in a graphical computing environment , as opposed to working from the command line, we recommend that you set up an NX connection. This provides you with a full graphical CloudBioLinux desktop. For a given instance that you have launched, you must go through the steps in the following two sections once. After that, you will be able to connect to a graphical desktop session for your launched instance as often as you like.

For those comfortable in text-only environments, including Linux users who wish to run graphical programs, without a full desktop, you need only follow the instructions in the next section one time. Then using the the instructions in the following section, you can log into a terminal using ssh as often as you like.

For Windows users who wish to have access to graphical programs, it is easiest to run an NX connection.

2.3.2 Find out the address of your instance

You need to know the address that's been assigned to your image, so that you can tell ssh or NX which machine you are trying to connect to.

- Assuming you have already clicked on the Instances link on the left side of your EC2 Dashboard, click on the **Instance Actions** button near the top of the Instances page. See figure 2.3.
- Choose **Connect**. A window will open containing directions about how to connect to your CloudBioLinux instance using ssh. *You need to make a couple of changes to the suggested connection instructions*, described below.

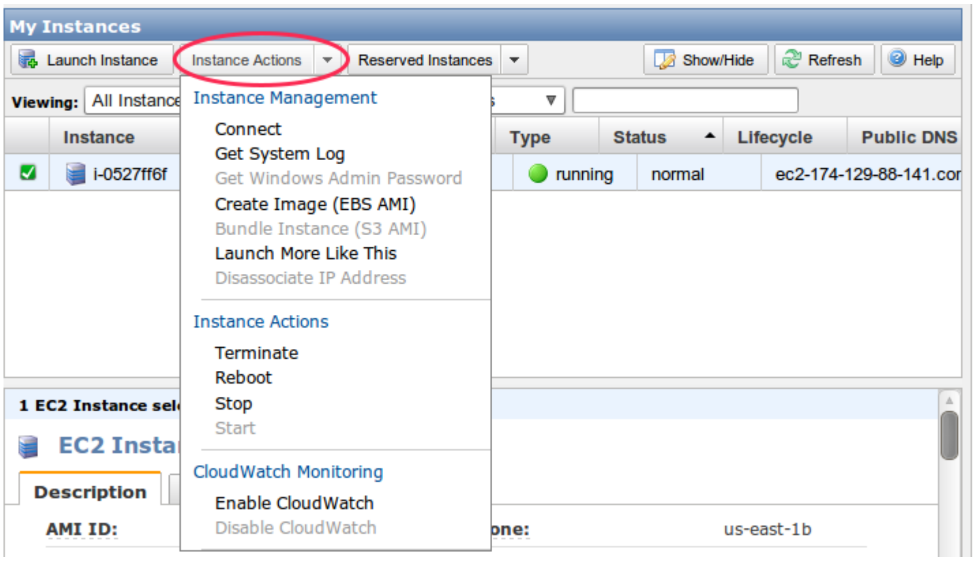


Figure 2.3: Click the Instance Actions button (pink circle) to bring up a menu with options including connecting to an instance you have already started up. This menu is also used to terminate a running instance.

2.3.3 Logging into a terminal using ssh

After you clicked on the Connect option to connect to your instance, you should have seen a small window pop up. The instructions in that window should have text similar to the following in it:

ssh -i mykey.pem root@ec2-184-72-144-209.compute-1.amazonaws.com

The text after the @ symbol is the address of your running instance. If you are working on Linux, or you have an ssh program on Windows with a terminal, the information in the window is *similar to* the command you could use to connect to your instance [2]. CloudBioLinux is based on Ubuntu. **To log into the instance, you need to use the *ubuntu* user**, not the root user[3]. So, an example command you might run in a terminal to connect to your instance is:

ssh -i mykey.pem ubuntu@ec2-184-72-144-209.compute-1.amazonaws.com

where you have used the ubuntu username (instead of root), and you include your machine address after the @ symbol.

[2] If you are logging in using Putty on Windows, you will need to enter the relevant information into the Putty system in order to connect.

[3] The default for most systems on Amazon EC2 is to log in as the root user.

Note: If you get a warning when you try to connect that suggests that your key cannot be found, it may mean that you have saved your key to a non-standard location and/or given it a non-standard name. In this case add path information for your key to the command line so that the private key can be found from where you run the ssh connection command. For example, if your key is stored in a subdirectory of your home directory called *keys*, and you want to log in as the *ubuntu* user, you could log in using ssh and the command, you need to

ssh -i /home/mydirectory/keys/mykey.pem ubuntu@ec2-184-72-144-209.compute-1.amazonaws.com

The first time you connect to your running CloudBioLinux instance, you should be offered the opportunity to set up NX on your instance. At this point, you can also provide a username other than "ubuntu", and this new user will be created for you.

You can continue at this point to run programs on the command line, or if you are working on a Linux system, you can launch graphical applications using the command line. However, we generally recommend connecting to your CloudBioLinux instance using an NX client instead of a text-based ssh client. This is because many people find a graphical desktop environment easier to work on, and the menus and desktop links help people take full advantage of the facilities of the CloudBioLinux system. To log in via NX, just follow the instructions in the next section. You may wish to log out of your current ssh session (although this is not necessary).

2.3.4 Logging into graphical desktop using NX

Start up your NX client software[4]

If you are using a Nomachine NX client, you should now see an NX connection wizard. Here, you need to enter the address of your launched instance (the same address you used to log into the terminal earlier), and you should change the desktop type to Gnome. See figure 2.5.

When you've logged in, you should see a desktop similar to that in figure 2.6.

2.3.5 The logic of the NX setup

It may initially seem strange that you need to log into a command terminal before you can log into your EC2 instance using an NX client. The reason

[4] Images shown here refer to the Nomachine NX client for Linux, running on a standard NEBC Bio-Linux machine, but the process should be similar no matter what type of system you are working on (Linux, Windows, Mac).

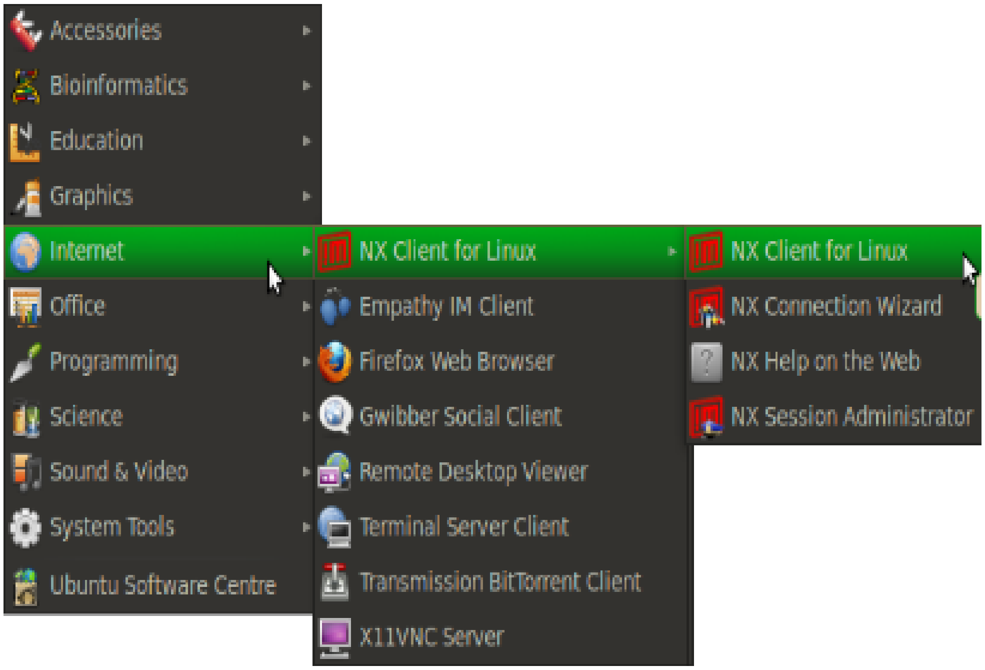


Figure 2.4: If an NX client is already installed, you should be able launch it from under your Applications menu on Ubuntu-based machines, or from your programs listing on Windows machines.

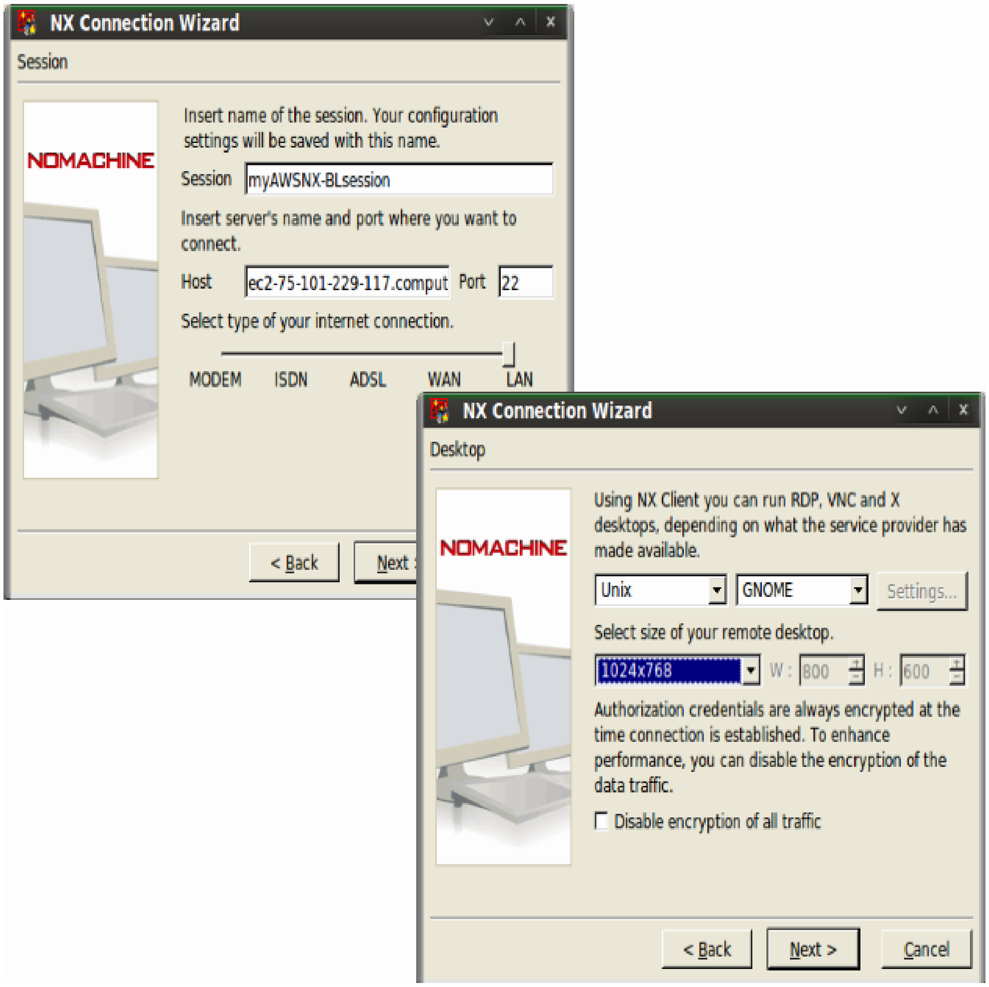
that this needs to be done is that key auth support, which is used for the ssh connections into the EC2 instances, is not supported by NX. Thus, for any given instance that is started, *you need a password to be able to log in using NX.*

When you set a password during your terminal session, you are ensuring that only you have that password. You also have the opportunity to create another user on the system, just by providing a username other than ”ubuntu” when prompted.

2.4 Logging out of your CloudBioLinux instance

From an NX connection you need to go to the System menu and choose the option **Shut Down...**. See figure 2.7.

From an ssh command line (or Putty) connection you need to type **exit** at the command prompt.



NX Connection Wizard

Session

Insert name of the session. Your configuration settings will be saved with this name.

Session: myAWSNX-BLsession

Insert server's name and port where you want to connect.

Host: ec2-75-101-229-117.comput Port: 22

Select type of your internet connection.

NX Connection Wizard

Desktop

Using NX Client you can run RDP, VNC and X desktops, depending on what the service provider has made available.

Unix GNOME Settings...

Select size of your remote desktop.

1024x768 W: 800 H: 600

Authorization credentials are always encrypted at the time connection is established. To enhance performance, you can disable the encryption of the data traffic.

□ Disable encryption of all traffic

Figure 2.5: Enter the same machine address you used for your terminal login earlier. Remember to use the same username you provided earlier - this may be the default username, ”ubuntu”, or it may be something else that you chose when you set the NX password. Please choose the gnome desktop type.

2.5 Terminating your CloudBioLinux instance

Highlight the instance you wish to terminate in the list on your Instances page. Click on the **Instance Actions** button (see figure 2.3) and choose **Terminate** under the Instance Action section of the menu. In basic terms, terminating results in the system and all the files and data on it being deleted [5]. If you have

[5] You will still be charged a fee if you have only stopped your instance, as opposed to terminating it, and your data may still be deleted depending on how you have set things up.

work you wish to save before terminating, or if you wish to keep a copy of this image so that you can use it later, without paying as much as you would for a running instance, please check out the Amazon documentation on EBS Volumes and taking snapshots of instances.

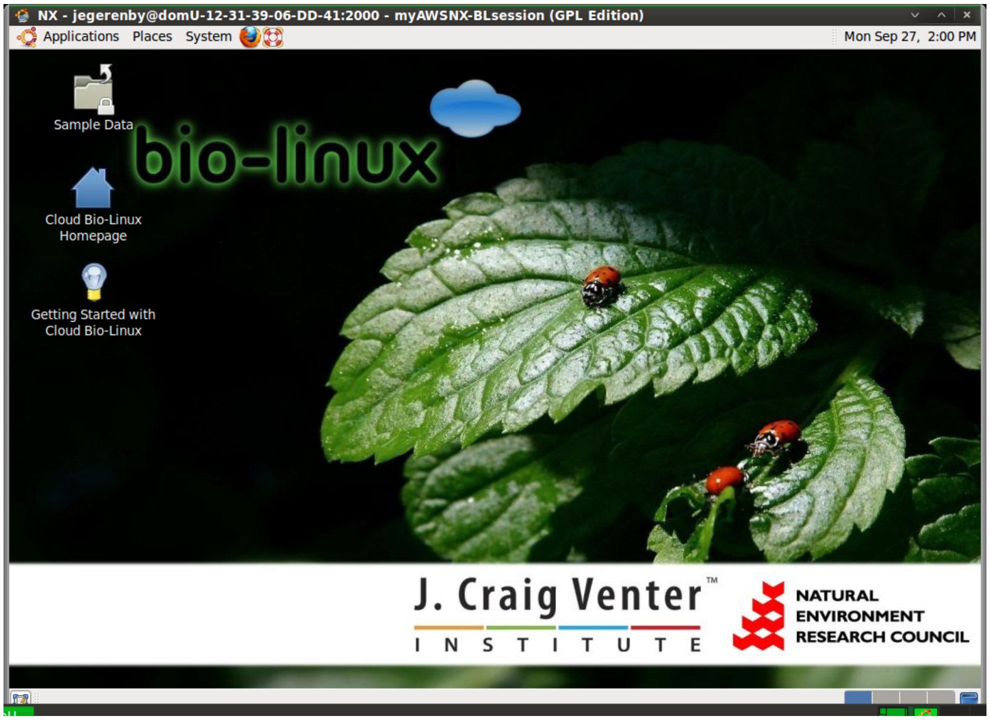

NX - jegerenby@domU-12-31-39-06-DD-41:2000 - myAWSNX-BLsession (GPL Edition)
Applications
Places
System
Mon Sep 27, 2:00 PM
Sample Data
Cloud Bio-Linux Homepage
Getting Started with Cloud Bio-Linux
bio-linux
J. Craig Venter™ INSTITUTE
NATURAL ENVIRONMENT RESEARCH COUNCIL

Figure 2.6: The CloudBioLinux desktop. It includes a number of menus, as well as icons for folders containing sample data relevant for the bioinformatics software installed, and a link to this document.

We provide a short introduction to the Cloud Bio-Linux desktop in Appendix A.2 on page .

Stopping and terminating are different.

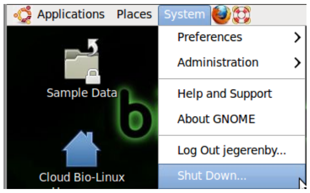


Figure 2.7: Choosing the **Shut Down...** option in an NX session logs you out. Logging out is not the same as terminating your CloudBioLinux instance. You will still be charged while the instance is running - whether you are logged into it or not.

Chapter 3

Working with data on the cloud

3.1 Introduction

For many bioinformatics tasks, you will want to work on your own data and files for example, perhaps your own sequence data and blast databases. To do this, you will need to *upload your files onto a machine that your CloudBioLinux instance can access.* Three options are covered in this chapter:

1. Copy your data directly onto the CloudBioLinux instance you are running. See section 3.2 This would be alright if you were going to use this data only on this running instance and you're happy for it to be deleted when you terminate the instance.

2. Copy your date onto a separate EBS volume. This would be useful if you wish to store your files for use in other sessions, but you do not plan to keep the same running instance. (EBS Volumes are cheaper than running instances.) See section 3.3

3. If the data you want to use is already available on Amazon EBS volumes (for example, ENSEMBL data), you can access this easily, with no data transfer costs. 3.4.

Snapshots are not discussed in this document, but are an important topic to understand if you plan to use the Amazon Cloud, especially if you will be working with your own data over a prolonged period or making customisations you do not wish to lose.

3.2 Copying data onto your CloudBioLinux instance

If you only need your data for a single CloudBioLinux instance, then you can just copy your data onto that instance directly. Once you are logged into your CloudBioLinux instance, there are a number of ways to do this. For example, there are command line tools like **scp**, for copying files from a machine you have an account on, or **wget** to bring in data from public websites or ftp sites.

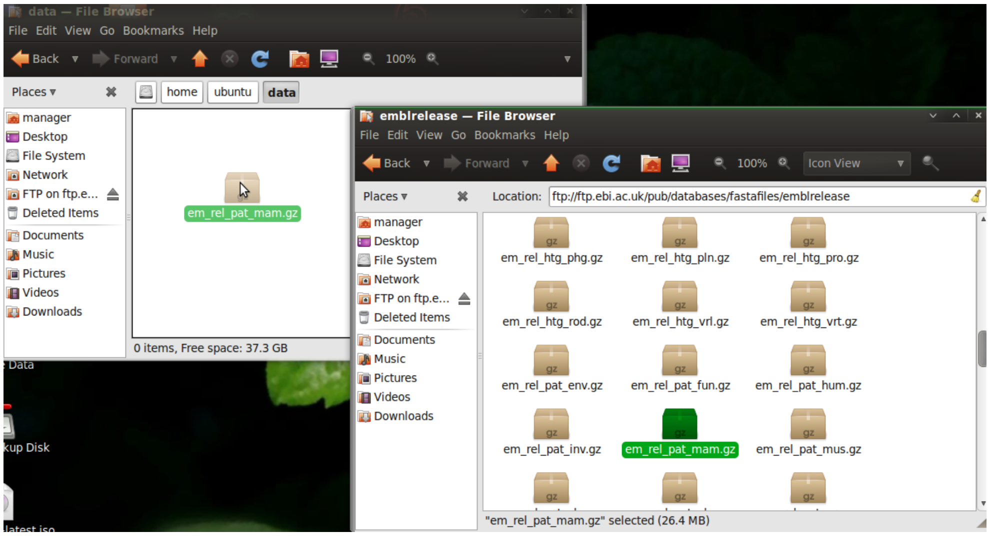
Alternatively, if you are logged into the full graphical desktop using NX (information on page 12), you can use the file browser to connect to a remote site and **drag and drop** your files to your running Bio-Linux instance. This is the method we focus on here.

- Go to the **Places** menu in the top taskbar and open up a file browser, for example by clicking on your Home Folder.

- Now go to the **Go** menu and click on Location... (or just type Ctrl-L).

- If you are going to copy files from a machine that you have login permissions on, then in the box next to the word *Location* that appears in your file browser, type: **ssh://your.machine.com**, replacing your.machine.com with the address of the machine your files are on. Alternatively, if you wanted to copy files from a public ftp server, say, then you would enter something like the following in the Location box:
  ftp://ftp.someother.database.site

As a specific example, if I want to copy fasta files from the EMBL database sections, I would type the following into the Location box:
ftp://ftp.ebi.ac.uk/pub/databases/fastafiles/emblrelease



0 items, Free space: 37.3 GB

Location: ftp://ftp.ebi.ac.uk/pub/databases/fastafiles/emblrelease

"em_rel_pat_mam.gz" selected (26.4 MB)

Figure 3.1: Copying files from remote machine is easy using the graphical File Browser, which can be launched from under the **Places** menu in the top taskbar. Choosing the Locations option under the Go menu of the file browser will allow you to type in a protocol (e.g. ftp, ssh) and a location. Here, a section of EMBL from the EBI is copied to my system using drag and drop between the two file browser windows.

Now open another file browser by going to the **Places** menu. Navigate to the folder you wish to store the files in. Now you can just drag and drop your files from the remote machine onto your CloudBioLinux instance. See figure 3.1 on page 19.

This process is simple, and for one-off jobs, is perfectly adequate. Note that you will generally pay for the network traffic you generate in transferring the data[1] So if you are going to use the same dataset numerous times, it is worth considering setting up an EBS volume rather than transferring data onto new instances. Even if this transfer is free, it will still generally take more time than mounting an EBS volume that already has your data on it.

3.3 Using EBS volumes for data

An Amazon EBS volume is what you need if

- you are going to use a dataset a number of times, with gaps in time between uses, or
- you want to store your data such that you can connect to it from different CloudBioLinux (or other Amazon EC2) images, or
- if you wish to share your data with other people also working on Amazon EC2 systems.

This guide presents only a small part of what is possible with EBS volumes. Please check out the EBS volume documentation on the Amazon website for further information.

A note on charging: You will be charged for your Amazon EBS volume as long as it is in existence, and you will be *charged for the space you request*, not the space you are really using. So if you ask for 1Gb, you are paying for 1Gb, even if you only use 100Kb.

Creating your volume

The instructions in this section assume you have already started up and have logged into an EC2 instance.

The steps involved in creating and mounting a volume are illustrated in figure 3.2.

- In the Navigation pane (left side) of the AWS Management Console, go to the Elastic Block Store area and choose **Volumes**. See figure 3.3.
- Click on the **Create Volume** button.
- Choose the same availability zone as the images you plan to use this volume with.

[1] Until November 1, 2010, data transfer onto Amazon is free. The first Gb per month of transfer off is also free. (Information taken from the Amazon website on July 21, 2010, with no guarantees to be correct at the time you are reading this document. Please check the official pricing list.

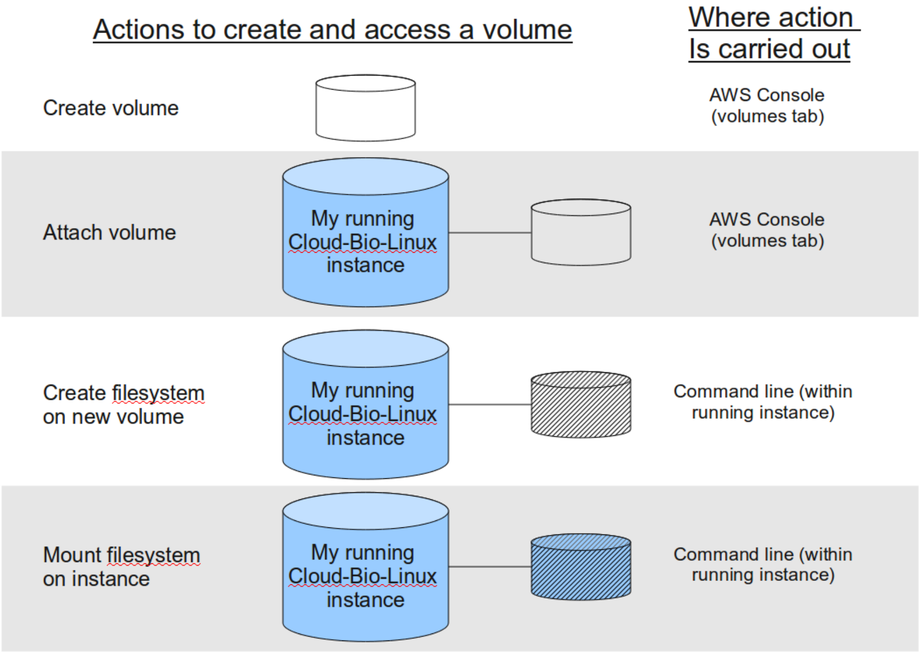
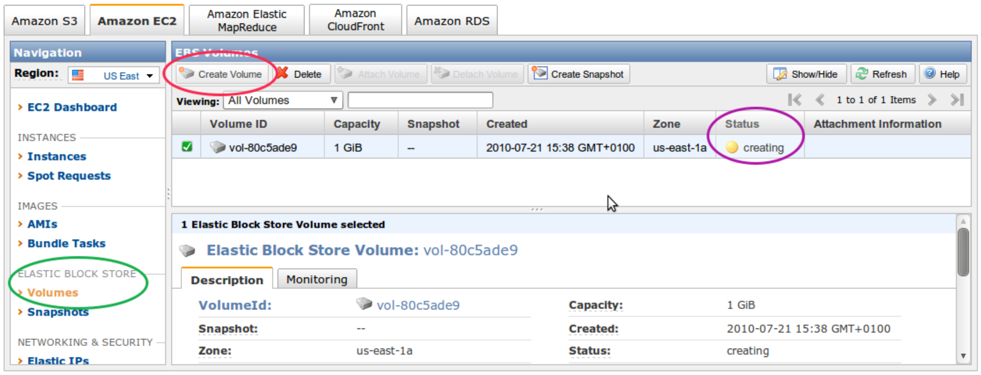

Actions to create and access a volume
Where action Is carried out
Create volume
AWS Console (volumes tab)
Attach volume
My running Cloud-Bio-Linux instance
AWS Console (volumes tab)
Create filesystem on new volume
My running Cloud-Bio-Linux instance
Command line (within running instance)
Mount filesystem on instance
My running Cloud-Bio-Linux instance
Command line (within running instance)

Figure 3.2: To create a new volume and get access to it from your running instance involves 4 steps. Here we suggest that the first two are done using the AWS Console. The second two steps are done within your running instance. If you want to work with a volume you (or someone else) has created earlier, then you need only carry out the steps in the grey boxes. The commands for carrying out the two steps in your running instance are provided in the text on page 22

- After changing any other settings in this window, press the **Create** button.
- Wait until the yellow circle beside the word *creating* is replaced by a blue circle beside the word *available*.

You now have an EBS volume attached to your running instance. This is analagous to plugging in a new piece of hardware to a server. This means that the volume **is not yet ready for copying data onto**. You first must mount the volume on your instance - this makes the volume accessible to you when you are logged into your instance [2].

[2] For those used to user-friendly plug 'n play systems, where you just plug a pre-formatted hard drive into a machine and copy to it directly, the steps outlined here might come as a bit of long-winded shock. In fact, all you are doing is carrying out the steps that, between your hardware vendor and your computer operating system, are often done for you. In particular, the equivalent to a disk drive here is the EBS volume, which is initally unformatted and needs



Viewing: All Volumes

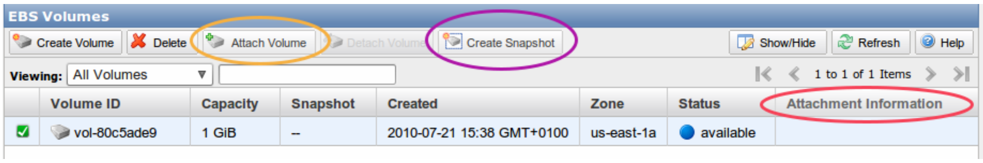
	Volume ID	Capacity	Snapshot	Created	Zone	Status	Attachment Information
☑	vol-80c5ade9	1 GiB	--	2010-07-21 15:38 GMT+0100	us-east-1a	● creating	

1 Elastic Block Store Volume selected

Elastic Block Store Volume: vol-80c5ade9

VolumeId:	vol-80c5ade9	Capacity:	1 GiB
Snapshot:	--	Created:	2010-07-21 15:38 GMT+0100
Zone:	us-east-1a	Status:	creating

Figure 3.3: The volumes pane of the AWS Console is brought up by clicking on the Volumes link in the Navigation pane (Green oval). Red circle: Just click the Create Volume button - it does what it says. Purple circle: the status of your volume creation is reported to you. When the circle turns blue and the word says *available*, you can proceed to attach and mount your volume.



Viewing: All Volumes

	Volume ID	Capacity	Snapshot	Created	Zone	Status	Attachment Information
☑	vol-80c5ade9	1 GiB	--	2010-07-21 15:38 GMT+0100	us-east-1a	● available	

Figure 3.4: Click on the Attach Volume button (Orange circle) to Attach a volume to an instance.

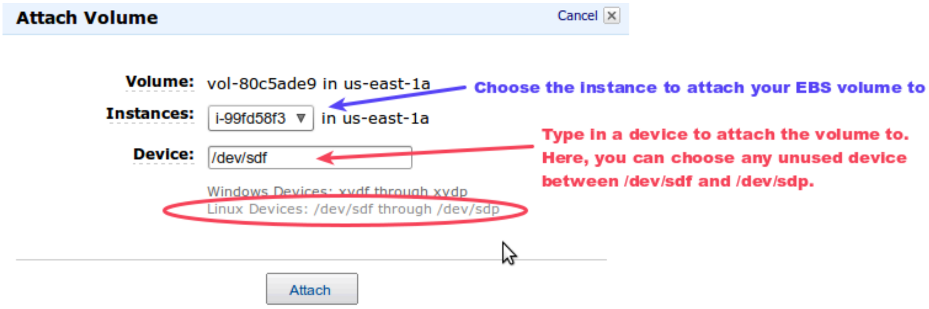
- Click on the **Attach Volume** button on the Volumes page. See figure 3.4.
- To attach a volume using the AWS Console interface, you just need to fill in the requested information. See figure 3.5.

Note that selecting any of your volumes in the AWS Management Console will bring up details of that volume at the bottom of the page.

Accessing your volume

This is where it gets a bit ugly, as you need to log into your machine and use the command line for the next couple of steps. The first of these steps,

to be formatted. To do this, you attach it to your running instance, then you format it. Now, to work with the filesystem on your volume, you need to mount in onto your image file system, so the whole thing - image and volume, can be interacted with as one. Until you unmount the volume of course.



Volume: vol-80c5ade9 in us-east-1a

Instances: i-99fd58f3 in us-east-1a

Device: /dev/sdf

Windows Devices: xvdf through xvdp

Linux Devices: /dev/sdf through /dev/sdp

Choose the instance to attach your EBS volume to

Type in a device to attach the volume to. Here, you can choose any unused device between /dev/sdf and /dev/sdp.

Figure 3.5: Fill in the information requested. Note that both the instance and the EBS volume are in the same region. You just need to provide an unused device to attach your volume to. If you don't know what this means, and are starting up a CloudBioLinux instance, then just use any of the suggested locations: /dev/sdf, /dev/sdg, /dev/sdh....up to /dev/sdp for now and check out some documentation to find out what it means later.

formatting your disk, only needs to be done *the first time* you use a particular volume. The second step, **mounting the volume**, needs to be done *each time* you want to access data on your volume from a new instance. There are ways to automate mounting a volume onto new instances, but these are not covered here.

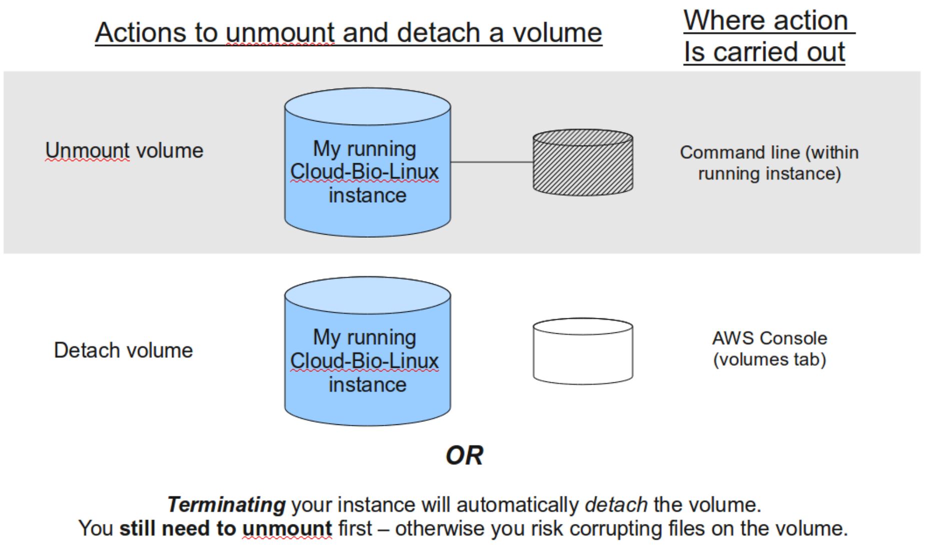
1. Log into your instance. If you are logged in using NX, start up a terminal window.

2. **The first time you mount a volume for use:** Type the following command to create an ext3 filesystem on your volume. Here I assume you have mounted it to /dev/sdf. See figure 3.5 on page 23.
   sudo mkfs -t ext3 /dev/sdf

3. Now make an empty directory. This will act as a place where you can mount your new filesystem. By convention, mount points tend to be put in directories such as /mnt or /media. For example, the command below creates a directory called /mnt/datasets:
   sudo mkdir /mnt/datasets

4. Now you can mount your volume onto this:
   sudo mount /dev/sdf /mnt/datasets

You will now be able to put data under the folder /mnt/datasets. All files under that directory are on your EBS volume and will not be lost when your instance terminates. Ensure you read the section on page 24 on **unmounting your volume** as failure to unmount before detaching your volume (or terminating your instance) could lead to data corruption.


Actions to unmount and detach a volume
Where action Is carried out
Unmount volume
My running Cloud-Bio-Linux instance
Command line (within running instance)
Detach volume
My running Cloud-Bio-Linux instance
AWS Console (volumes tab)
OR

Terminating your instance will automatically *detach* the volume.
You **still need to unmount** first – otherwise you risk corrupting files on the volume.

Figure 3.6: You must unmount a volume before detaching it. Not doing so may result in data corruption on the volume. You can then either detach the volume from your instance by using the **Detach Volume** button on the AWS Console, or the volume will be detached for you when you terminate your instance. The commands for unmounting a volume from within your running instance are provided in the text on page 24

Unmounting your volume

This is a simple but vital step to avoid the possibility of data corruption. Do this *before you detach* your volume or terminate your instance.

If you had attached your device to /dev/sdf, then you simply need to type:
umount -d /dev/sdf

Detaching your volume

You can detach your volume from your instance using the AWS Management Console using the Detach Volume button on the console Navigation pane. Alternatively, your volumes will be detached automatically when you terminate your instance.

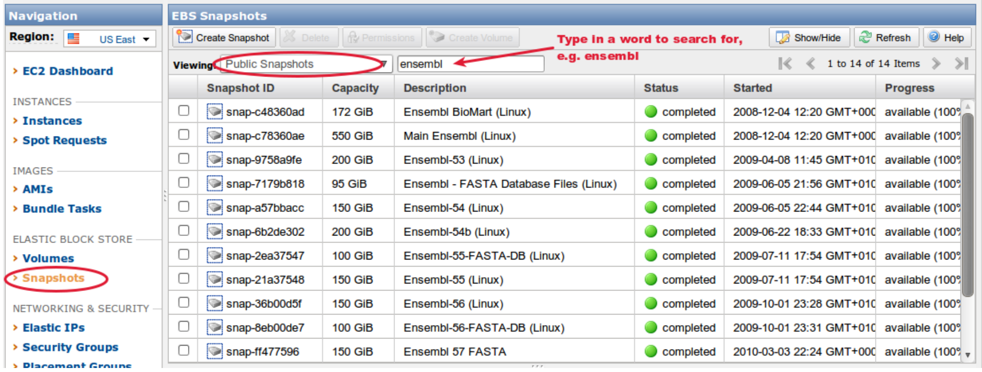
Backing up or sharing your volume

Check out the Userguide information on creating snapshots and on modifying permisisons on snapshots.

Deleting your volume

You can delete your volume using the Delete button on the console Navigation pane.

3.4 Accesing public datasets on Amazon



	Snapshot ID	Capacity	Description	Status	Started	Progress
□	snap-c48360ad	172 GiB	Ensembl BioMart (Linux)	● completed	2008-12-04 12:20 GMT+000	available (100%
□	snap-c78360ae	550 GiB	Main Ensembl (Linux)	● completed	2008-12-04 12:20 GMT+000	available (100%
□	snap-9758a9fe	200 GiB	Ensembl-53 (Linux)	● completed	2009-04-08 11:45 GMT+010	available (100%
□	snap-7179b818	95 GiB	Ensembl - FASTA Database Files (Linux)	● completed	2009-06-05 21:56 GMT+010	available (100%
□	snap-a57bbacc	150 GiB	Ensembl-54 (Linux)	● completed	2009-06-05 22:44 GMT+010	available (100%
□	snap-6b2de302	200 GiB	Ensembl-54b (Linux)	● completed	2009-06-22 18:33 GMT+010	available (100%
□	snap-2ea37547	100 GiB	Ensembl-55-FASTA-DB (Linux)	● completed	2009-07-11 17:54 GMT+010	available (100%
□	snap-21a37548	150 GiB	Ensembl-55 (Linux)	● completed	2009-07-11 17:54 GMT+010	available (100%
□	snap-36b00d5f	150 GiB	Ensembl-56 (Linux)	● completed	2009-10-01 23:28 GMT+010	available (100%
□	snap-8eb00de7	100 GiB	Ensembl-56-FASTA-DB (Linux)	● completed	2009-10-01 23:31 GMT+010	available (100%
□	snap-ff477596	150 GiB	Ensembl 57 FASTA	● completed	2010-03-03 22:24 GMT+000	available (100%

Figure 3.7: Click the Snapshot link in the Navigation pane. Then select Public Snapshots in the Viewing menu. Searching for a term such as *ensembl* brings up all the public snapshots that contain ensembl in their description.

Amazon makes some public data sets available as snapshots. You can just attach and mount these (see grey boxes in figure 3.2. No data transfer is necessary. Finding datasets is easy: just search through the public snapshots for relevant terms. See figure 3.7. Check out the full public data set listing.

Amazon provides documentation on how to make use of these public data resources.

Appendix A

Appendices

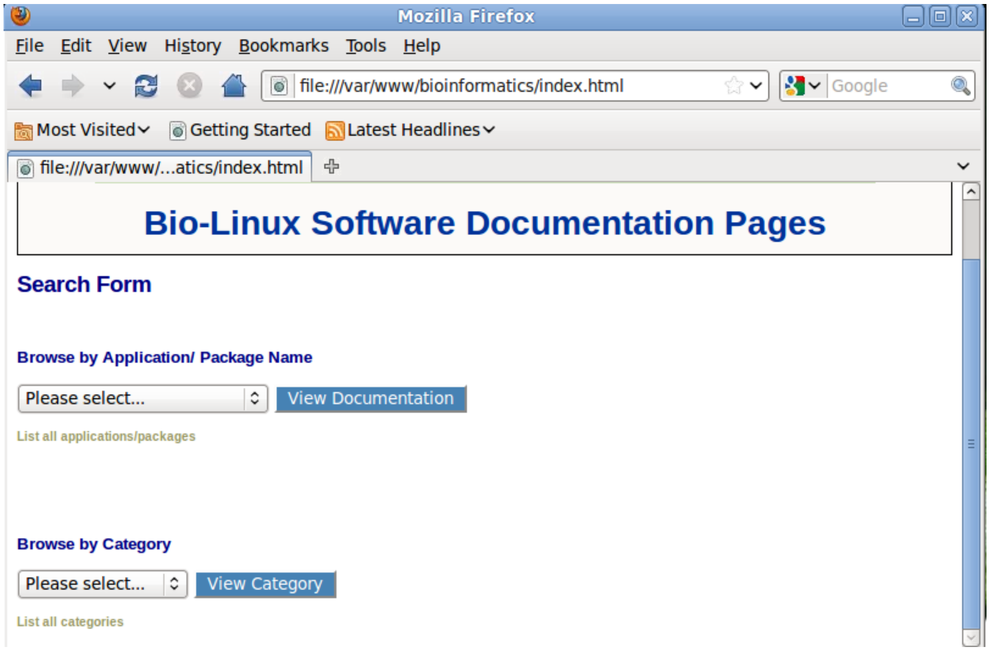
A.1 The CloudBioLinux Desktop

This section provides only a few tips about the CloudBioLinux desktop. As CloudBioLinux is based on Ubuntu, we recommend referring to online Ubuntu documentation for further details about the system in general.

A.1.1 Bioinformatics documentation

CloudBioLinux comes with a categorised bioinformatics documentation system, which provides information about the bioinformatics software on the system. This is the easiest place to find out about what bioinformatics programs are available, and where to find out more about how to use them.

At the moment, the easiest way to get to the installed documents is to open up a web browser (just click on the little firefox icon in the top taskbar), and then enter the address: **file:///var/www/bioinformatics/index.html**. See figure A.1.



file:///var/www/bioinformatics/index.html

Bio-Linux Software Documentation Pages

Search Form

Browse by Application/ Package Name

List all applications/packages

Browse by Category

List all categories

Figure A.1: A categorised bioinformatics documentation system. Perusing the information in these pages is probably the easiest way to start finding out what is available on your CloudBioLinux image.

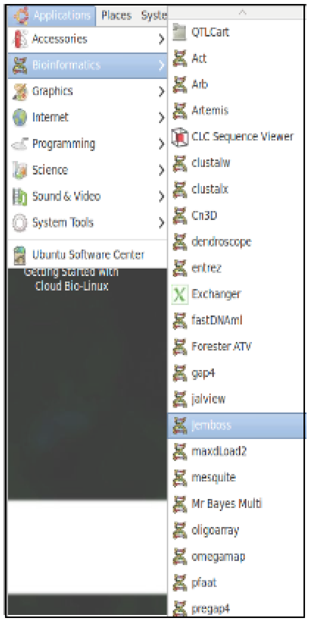


Figure A.2: Some bioinformatics programs can be launched from the Bioinformatics menu. For a more comprehensive listing of the bioinformatics software on the system, check out the installed bioinformatics documentation pages. See figure A.1.

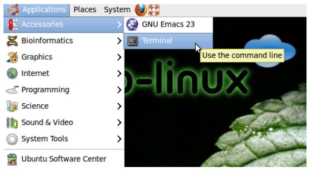


Figure A.3: You can open a command line terminal from under the Accessories submenu of the Applications menu in the top taskbar.

A.1.2 The bioinformatics menu

Some of the bioinformatics software on the system can be accessed via the graphical bioinformatics menu. See figure A.2.

A.1.3 Opening a terminal

Much of the bioinformatics software on CloudBioLinux systems can only be launched from the command line. This includes command line tools, and also some graphical software. In addition, much of the power of Linux comes through using the command line. You can easily open a terminal, in which you can type commands, when logged into a full desktop session. Just go to the **Accessories submenu** under the Applications menu in the top taskbar, and choose **Terminal**. See figure A.3.

A.2 Basic terminology

One of the most daunting parts of starting to work on the cloud is the abundance of new words and acronyms that litter the documentation. Here are a few of the most common terms you'll come across, and what they mean when used in the cloud computing context.

Amazon Machine Image - Pre-configured machine images that you can start up and work on. Most commonly referred to as AMI, or just "image" in documentation.

Amazon Simple Storage Service - More commonly known as "S3", this is a storage type offered as a web service. This document is primarily concerned with EBS Volumes for storage, rather than S3. However, if you take a snapshot of a volume, it is stored in S3. There are many differences between EBS and S3, which you should find out about if you plan to work on the cloud seriously.

AMI - see "Amazon Machine Image".

Availability Regions and Zones - Amazon have a number of data centres around the world. A Region in this context described centres in different geographic areas (e.g. the U.S. and Europe). An Availability Zone describes distinct locations within a region. For a new user, the most likely thing to look out for is that you create volumes in the same region and zone as your running instance(s). For more detailed information about regions and zones, check out the EC2 Userguide, including the Region and Availability Zone FAQ.

AWS - Stands for "Amazon Web Services". An umbrella name for the myriad of web services offered by Amazon. These services include the provision of compute power and storage, which are the focus of this document.

AWS console - A graphical web interface to your AWS account. From this interface, you can create, manage and delete Amazon resources such as system instances and data volumes.

EBS - See "Elastic Block Storage".

EC2 - See "Elastic Compute Cloud"

Elastic Block Strorage - Also referred to as "volumes". A type of storage that was designed for Amazon EC2 instances. You can create EBS volumes and mount these as devices, as you might with a hard drive on a standard system. EBS volumes are particularly useful if you plan to store data for multiple uses.

Elastic Compute Cloud - Also known as EC2, this is an Amazon web service that makes computational capacity available as a virtual computing environment. When you start up an image, you do so from within an EC2 account, from the EC2 section of the AWS Console.

Image - See "Amazon Machine Image"

Instance - See "Amazon Machine Image"

NX - At a user level, NX is a method allowing you to experience a full, graphical desktop session on a machine you are logged into remotely. Documentation outlining how this is achieved is available.

Putty - In the context of this document, Putty is a program that can be used on Windows as an SSH client.

S3 - See "Amazon Simple Storage Service"

SSH - Stands for Secure Shell. A network protocol that allows data to be exchanged using a secure channel. Information sent over an SSH connection is encrypted. The standard port for SSH connections is port 22.

Snapshot - A snapshot is essentially a copy of a volume. Taking a snapshot of an EBS volume means that you save the state of your volume, as a snapshot, in Amazon S3. This snapshot is then replicated across multiple Availability Zones. This means that the information on your EBS volume is saved in a durable manner. Snapshots can then be used as the starting point for new EBS volumes. For example, if you select a snapshot in the AWS Console, you can choose to make a volume of it. Once you have attached and mounted that volume to a running image, you have access to all the information you saved when you took the snapshot. Snapshots can also be used when sharing data or images. See the offical documentation for more.

Volume - See "Elastic Block Storage".