
labibi Documentation

Release 0.1

C. Titus Brown

May 23, 2013

CONTENTS

1	Start up an EC2 instance	1
1.1	The launch wizard	1
1.2	“Create a new instance” page 1	2
1.3	“Create a new instance” page 2	2
1.4	Wait for your instance to be running	2
2	Logging into your new instance “in the cloud” (Windows version)	5
2.1	Generate a ppk file from your pem file	5
2.2	Logging into your EC2 instance with Putty	8
3	Storing data persistently with Amazon	13
3.1	Prerequisites	13
3.2	Ask Amazon to create a new Elastic Block Storage Volume for you	13
3.3	Shutting down your instance	16
3.4	Snapshotting your volume	16
4	An Assembly Exercise	19
4.1	Start up an EC2 instance and log in	19
4.2	Install the ‘Velvet’ assembler	19
4.3	Grab some data	19
4.4	Assembling the data	20
4.5	Running multiple assemblies	21
4.6	Finishing up for today	21
4.7	Questions and thoughts to address	21
4.8	Reading	22
5	Indices and tables	23

START UP AN EC2 INSTANCE

Go to '<https://titus-courses.signin.aws.amazon.com>' in a Web browser.

Select 'My Account/Console' menu option 'AWS Management Console.'

Log in with username 'srop-student' and the password that the instructors give you.

Click on EC2 (upper left).

Select "Launch Instance" (midway down the page), and select "Quick Launch Wizard".

1.1 The launch wizard

Create a New Instance [Cancel]

Select an option below:

- ☐ **Classic Wizard**
Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.
- ☒ **Quick Launch Wizard**
Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.
- ☐ **AWS Marketplace**
AWS Marketplace is an online store where you can find and buy software that runs on AWS. Launch with 1-Click and pay by the hour.

Name Your Instance: Adam [Pick a meaningful name, e.g. Web Server]

Choose a Key Pair:
Public/private key pairs allow you to securely connect to your instance after it launches.

☐ Select Existing ☒ Create New ☐ None

Name: Adam [Download]

Please note that you need to download the key pair before you can continue.

Choose a Launch Configuration:

More Amazon Machine Images [New!]
Search through public and AWS Marketplace AMIs or choose from your own custom AMIs.

- Amazon Linux AMI 2013.03.1**
The Amazon Linux AMI is an EBS-backed, PV-GRUB image. It includes Linux 3.4, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat. 64 bit 32 bit [Free tier eligible]
- Red Hat Enterprise Linux 6.4**
Red Hat Enterprise Linux version 6.4, EBS-boot. 64 bit 32 bit
- SUSE Linux Enterprise Server 11**
SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS boot with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, and Ruby 1.8.7 available. 64 bit 32 bit
- Ubuntu Server 12.04.2 LTS**
Ubuntu Server 12.04.2 LTS with support available from Canonical (<http://www.ubuntu.com/cloud/services>). 64 bit 32 bit [Free tier eligible]

Note: You can customize your settings in the next step. [Continue]

Submit Feedback Getting Started Guide

On this page,

1. Name your new computer something (here, "Adam"; name it after yourself instead).
2. Create a new key pair (here, "Adam"; name it after yourself instead) and Download it.
3. Select "More Amazon machine images."

4. Click on “Continue.” This will be greyed out until you download the key pair (button, upper right).

Note: You only need to create a new key pair the first time you’re doing this – you can select the one you created the first time, if you still have a copy of the key file you downloaded stored somewhere.

1.2 “Create a new instance” page 1

Enter ‘ami-999d49f0’ into the search box and click “search”. You should see “starcluster-base-ubuntu-”. Select it, and hit Continue.

1.3 “Create a new instance” page 2

On this page, “Edit details” until it looks like the below image –

Create a New Instance Cancel

starcluster-base-ubuntu-11.10-x86_64 (ami-999d49f0)
Platform: Ubuntu StarCluster Base Ubuntu 11.10 x86_64 (Us-east-1)
Architecture: x86_64

Please review your settings and click **Launch** to finish or **Edit details** to make changes.

Instance Details

Name: Adam	Type: m1.large
Detailed Monitoring: No	Availability Zone: us-east-1c
Shutdown Behaviour: Stop	Termination Protection: No
Launch into a VPC: No	

Security Details

Key Pair: Adam	Security Group: default
----------------	--------------------------------

Advanced Details

Kernel ID: Default	Ramdisk ID: Default
User Data:	IAM Role:
Network Interfaces:	

[Go Back](#) [Edit details](#) [Launch](#)

1. Make sure your “Type” is m1.large.
2. Make sure your “Availability zone” is something specific, like us-east-1c.
3. Make sure your “Security group” is set to default.

Then, click “Launch”.

1.4 Wait for your instance to be running

Go to the ‘instances’ list and make sure your particular instance is running.

EC2 Dashboard

Launch Instance Actions

Viewing: All Instances All Instance Types Search

Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status	Monitoring	Security Groups	Key Pair Name	View
Elijah	i-d6d1febd	ami-999d49f0	ebs	m1.large	terminated	initializing...	Loading...	basic	default	eli	ps
Adam	i-f6897a93	ami-999d49f0	ebs	m1.large	running	initializing...	Loading...	basic	default	Adam	ps

EC2 Instance: Adam (i-f6897a93)

ec2-50-17-70-145.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

AMI: starcluster-base-ubuntu-11.10-x86_64 (ami-999d49f0)

Zone: us-east-1c

Alarm Status: Loading...

Security Groups: default. view rules

You'll need the hostname of your new computer, on the bottom (ec2-...) – we suggest selecting this and copying it somewhere.

Then, go to *Logging into your new instance “in the cloud” (Windows version)*.

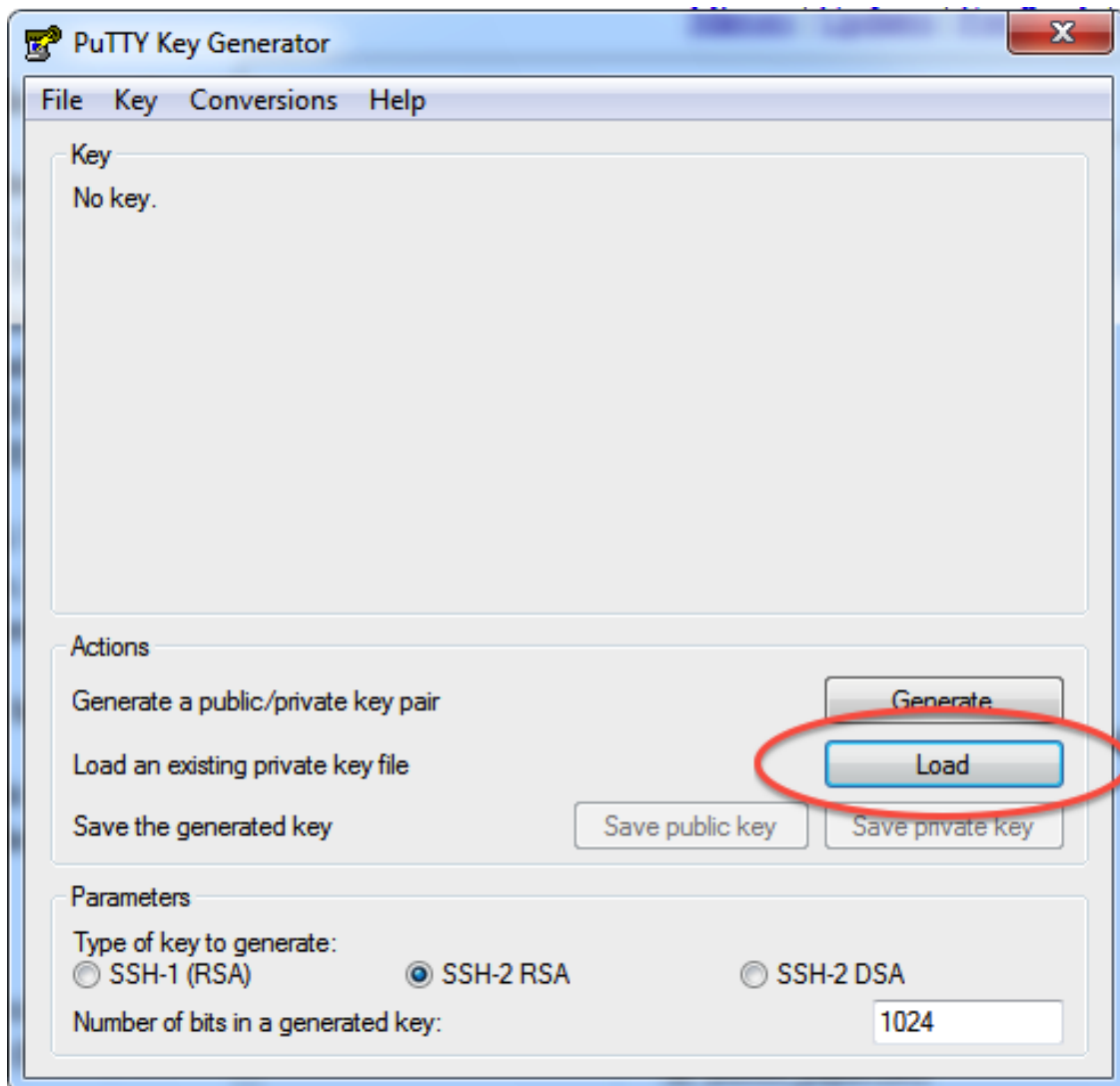
LOGGING INTO YOUR NEW INSTANCE “IN THE CLOUD” (WINDOWS VERSION)

Download Putty and Puttygen from here: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

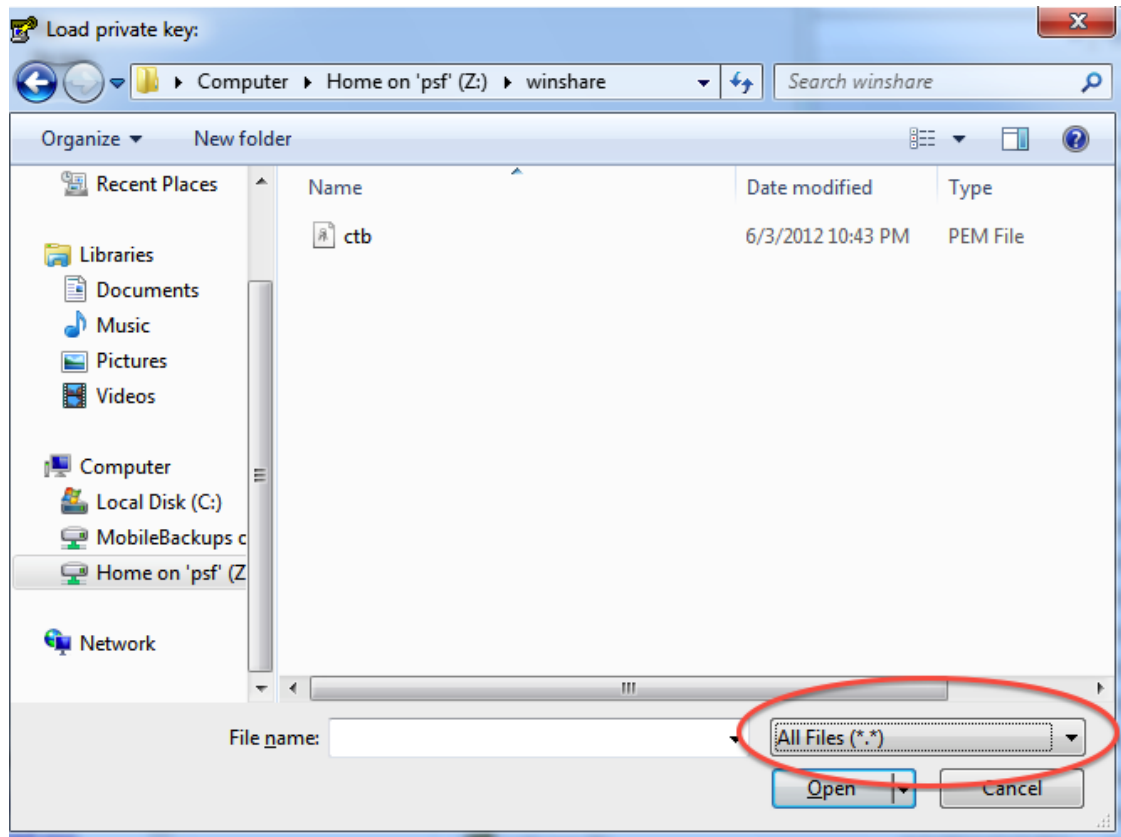
2.1 Generate a ppk file from your pem file

(You only need to do this once!)

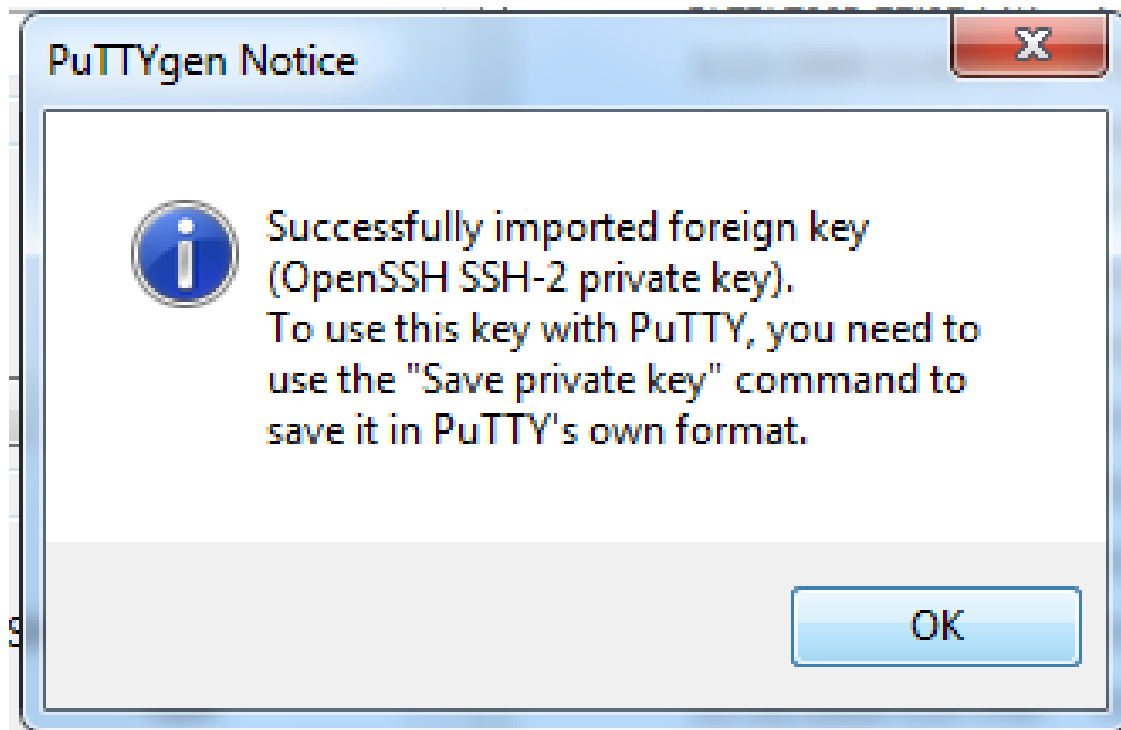
Open puttygen; select “Load”.



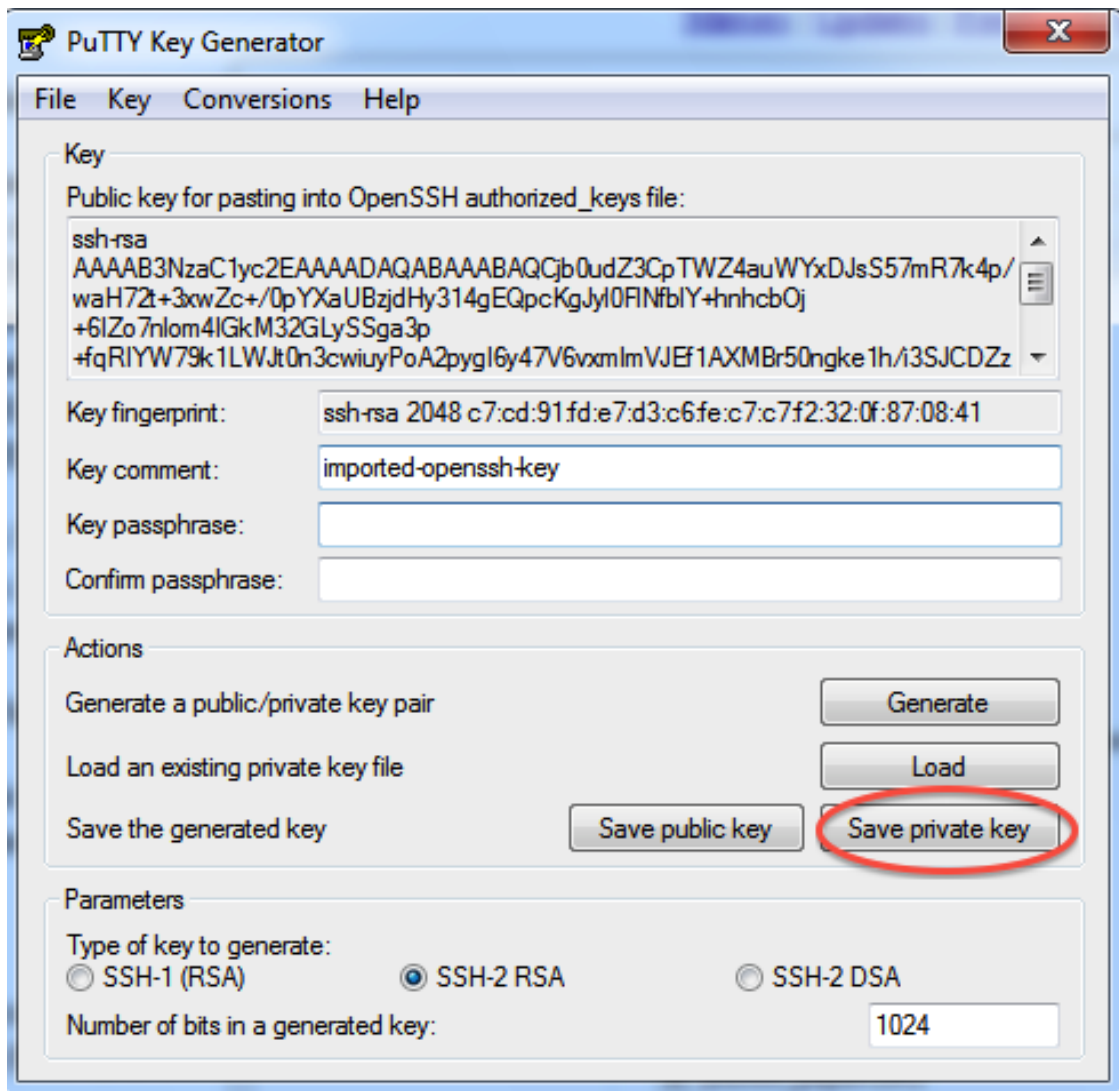
Find and load your '.pem' file; it's probably in your Downloads folder. Note, you have to select 'All files' on the bottom.



Load it.

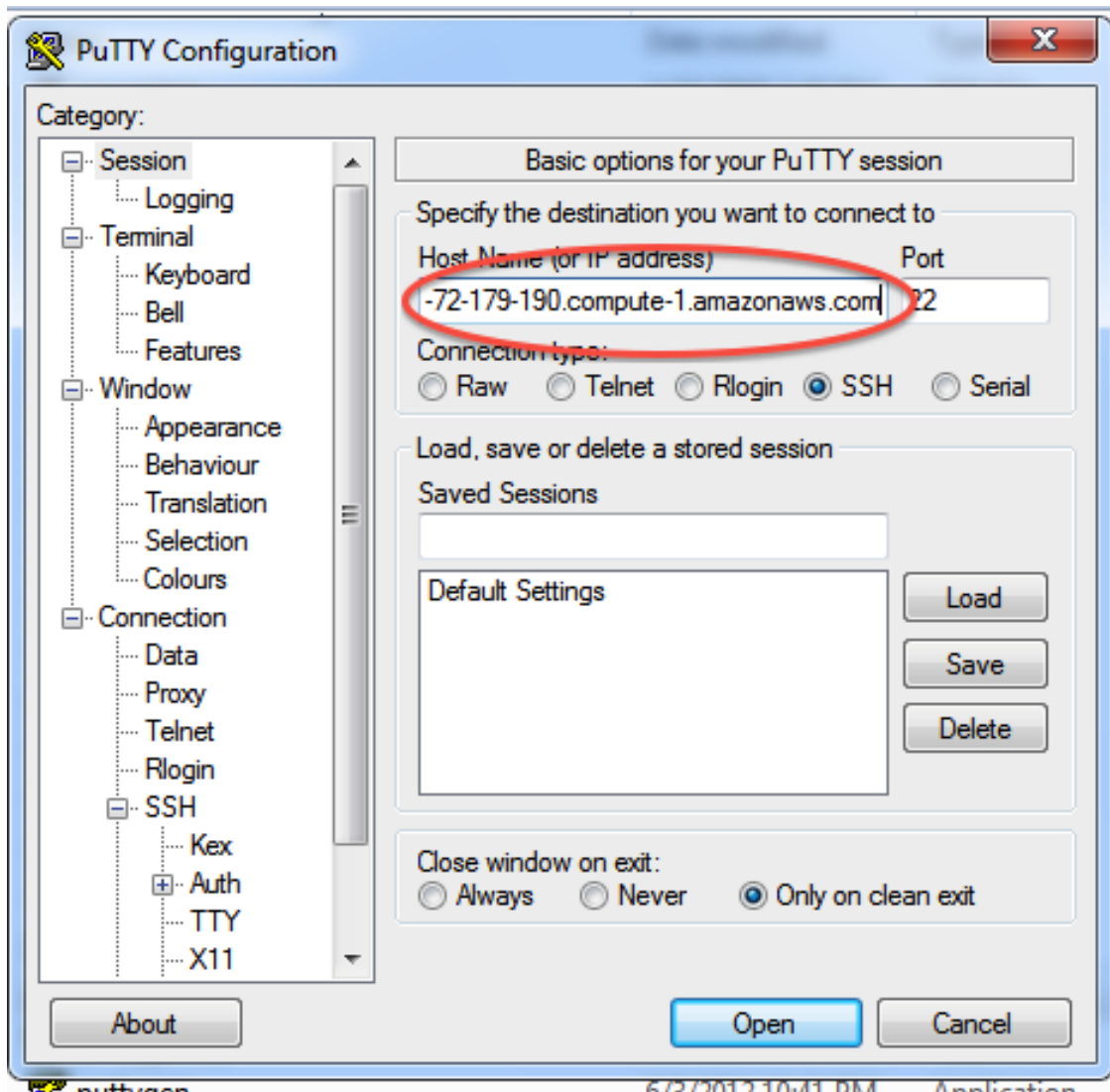


Now, "save private key". Put it somewhere easy to find.

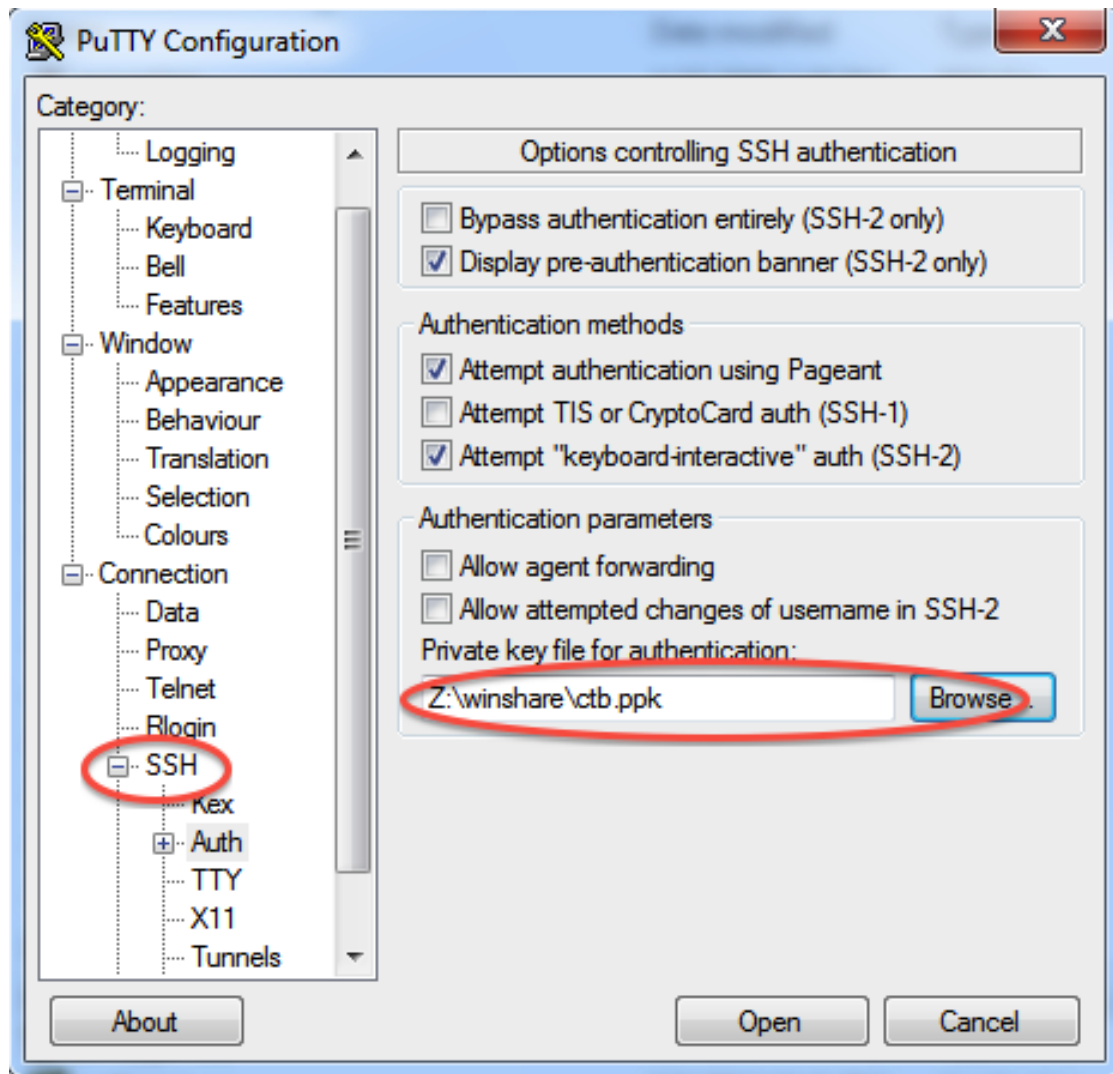


2.2 Logging into your EC2 instance with Putty

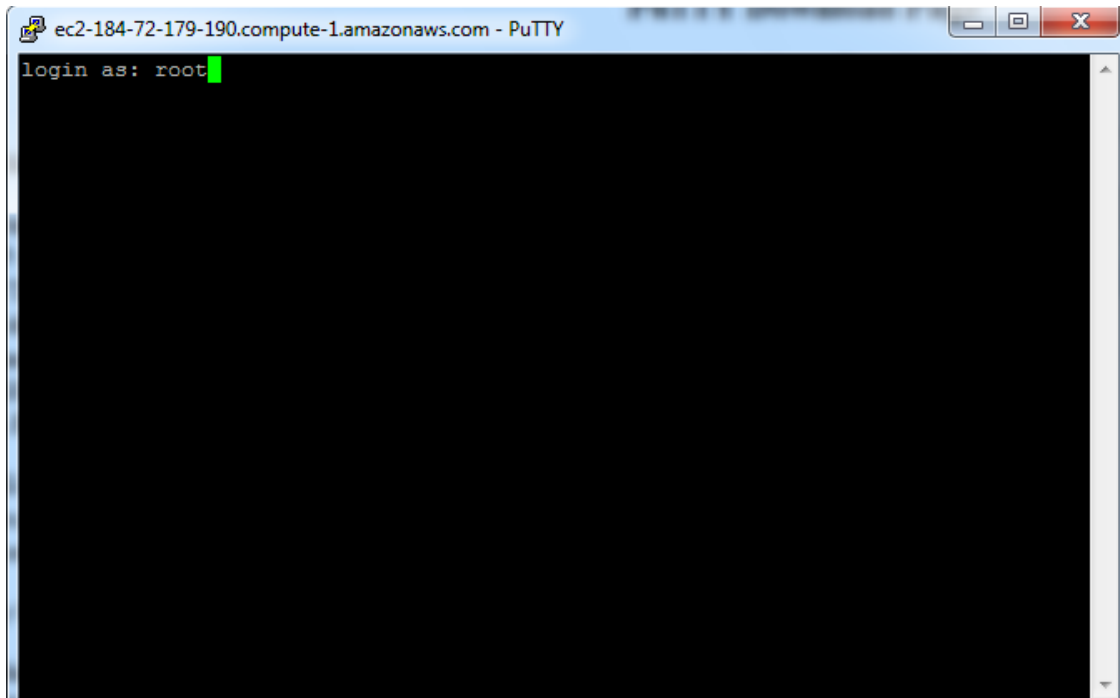
Open up putty, and enter your hostname into the Host Name box.



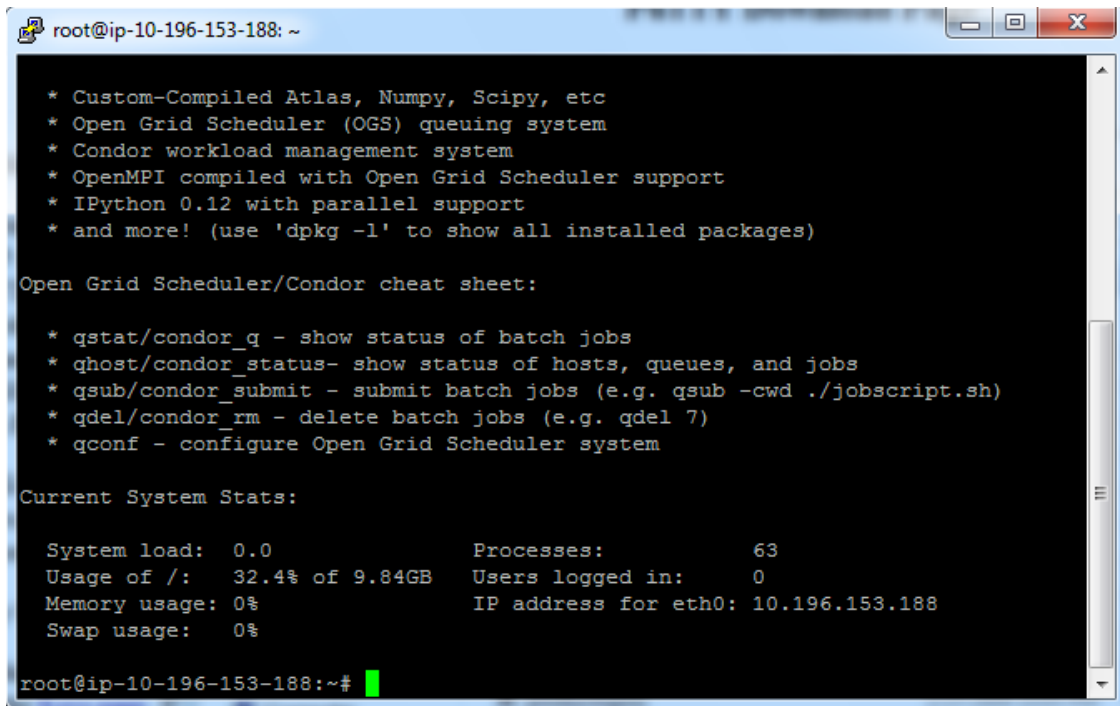
Now, go find the 'SSH' section and enter your ppk file (generated above by puttygen). Then select 'Open'.



Log in as “root”.



Declare victory!



STORING DATA PERSISTENTLY WITH AMAZON

Author Rosangela Canino-Koning and Titus Brown

Date May 21, 2013

If you want to save your data across instances – that is, if you want to have persistent data – Amazon can do that for you, too. You need to use the Amazon Elastic Block Storage service, which creates a virtual hard drive that you can (virtually) attach to your EC2 instance.

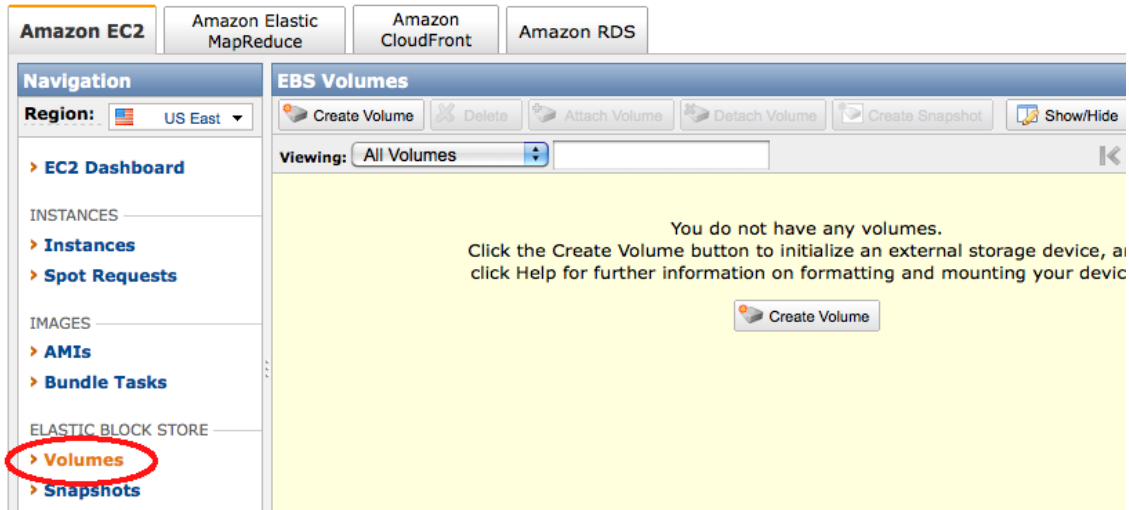
You can read more about EBS [here](#).

3.1 Prerequisites

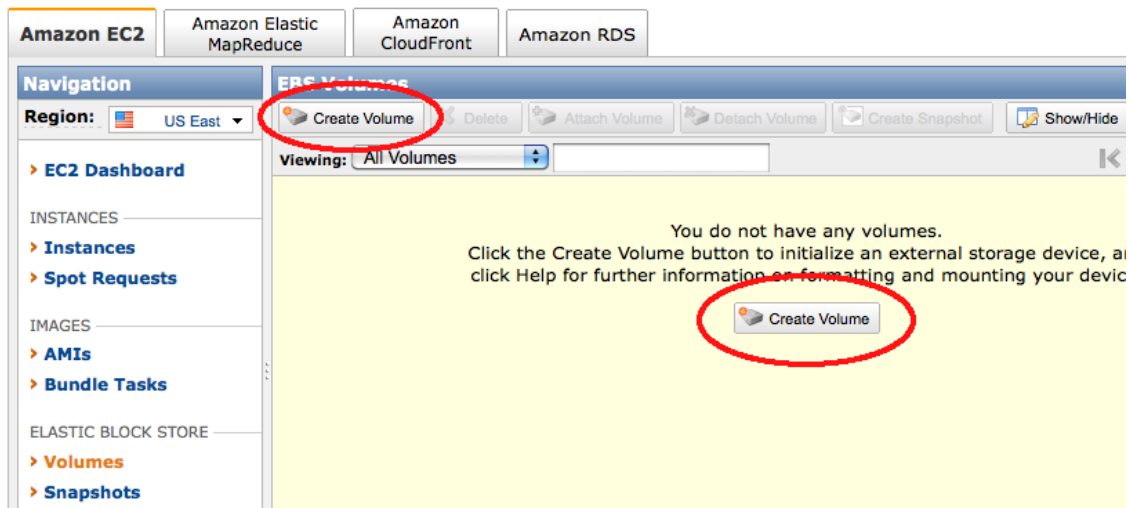
This tutorial assumes you’ve already got account on Amazon Web Services, and that you’ve completed the EC2 tutorial to set up an Amazon instance.

3.2 Ask Amazon to create a new Elastic Block Storage Volume for you

At the AWS Management Console, on the left menu bar, click “Volumes”.



Click “Create Volume”.



Enter the desired size, and select the zone in which your instance is running. **The volume and instance must be in the same zone.** Otherwise, the volume cannot be attached to your instance.

Then click “Create”.

Create Volume Cancel

Size: 20 GiB

Availability Zone: us-east-1a

Snapshot: --- No Snapshot ---

Create

Wait for your volume to finish being created, then click “Attach Volume”.

Amazon Elastic MapReduce Amazon CloudFront Amazon RDS

EBS Volumes

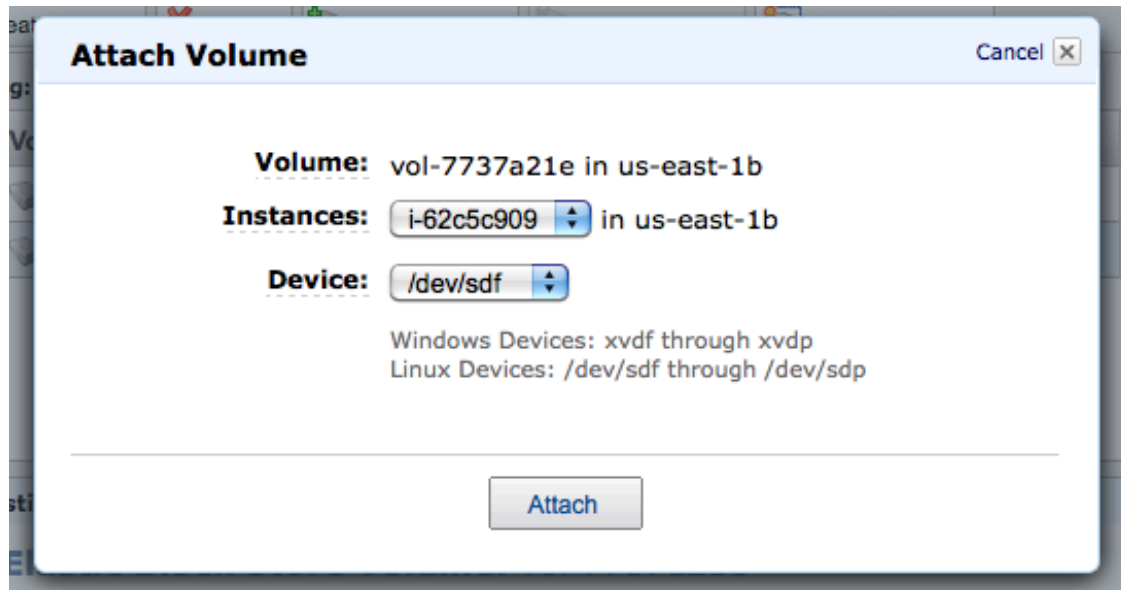
Create Volume Delete Attach Volume Detach Volume Create Snapshot

Viewing: All Volumes

	Volume ID	Capacity	Snapshot	Created	Zone
<input checked="" type="checkbox"/>	vol-8134a1e8	20 GiB	--	2010-06-08 14:53 EST	us-east-1a

Select the desired running instance. It will ask you for a device name to attach; this should be `/dev/sdf`; if you attach more than one, you can use `/dev/sdg`, etc. You can name them anything up to at least ‘i’ or ‘j’. Remember this for later – it’s how the computer will know which disk to “talk” to!

Click “Attach”.



When attachment is complete, connect to your instances via SSH.

If the volume is newly created, you must format the volume. **WARNING: ONLY DO THIS ONCE, WHEN YOU FIRST CREATE THE VOLUME. OTHERWISE, YOU WILL LOSE ALL YOUR DATA.**

```
mkfs -t ext2 /dev/xvdf
```

(If you used 'sdg' above, make it 'xvdg' etc. I know it's confusing, but that's just how computers work sometimes.)

It will ask you if you want to use the entire device – say “y” for “yes.

Then, mount the volume. You'll do this every time you attach the volume to an instance:

```
mkdir /work
mount /dev/xvdf /work
```

Your drive is now ready to use – it will be available under /work. Files copied into that directory or directories underneath it will be stored on your EBS volume.

3.3 Shutting down your instance

Any volumes you have attached will automatically detach when you shut down the instance. You can also stop all processes that are using the volume, change out of the directory, and type

```
cd
umount /work
```

and then detach the volume via the AWS Web site.

3.4 Snapshotting your volume

Snapshots are backups of your volume that you can share with other people. Snapshots are much more reliable long-term than volumes are, and you can use them as a basis for creating a new volume (in which case the new volume will start out containing all the data in the snapshot). So, if you upload some raw data and want to work with it over a few weeks, we suggest:

- create a volume and load the data onto the volume
- snapshot the original volume
- make a new volume from the snapshot, and delete the original volume

AN ASSEMBLY EXERCISE

Author

3. Titus Brown

Date May 22, 2013

4.1 Start up an EC2 instance and log in

Follow the instructions from yesterday (in *Start up an EC2 instance*) BUT with one modification: **use the machine image ‘ami-c17ec8a8’, instead of the other ami.**

Log in to the machine with SSH (as in *Logging into your new instance “in the cloud” (Windows version)*). (If you’re using a Mac, read `log-in-with-ssh-mac`.)

4.2 Install the ‘Velvet’ assembler

At the command prompt, copy and paste the following:

```
cd /root
curl -O http://www.ebi.ac.uk/~zerbino/velvet/velvet_1.2.08.tgz
tar xzf velvet_1.2.08.tgz
cd velvet
make MAXKMERLENGTH=51
cp velvet? /usr/local/bin
```

4.3 Grab some data

Now, let’s grab some read sequencing to assemble:

```
cd /mnt
curl -O https://s3.amazonaws.com/public.ged.msu.edu/ecoli-reads-5m-dn-paired.fa.gz
```

This takes some data that I’ve prepared for you, and downloads it into the file ‘ecoli-reads-5m-dn-paired.fa.gz’.

Let’s take a quick look at the contents here:

```
gunzip -c ecoli*.gz | head
```

This command uncompresses the data into text, and then shows you the first 10 lines. You should see this:

```
>EAS20_8_6_1_6_1407/1
AGCGATTGCTGTGTGGCTTCAACCACTGGACAGGAGCGCGTTTTTCAGGCTGATCACCAGTGCCTCGATTG
>EAS20_8_6_1_6_1407/2
TCTACCGGGAGCGAAATCATGATCAAGATTGGCGTTATCGCCGATGATTTTACCGGCGCGACGGATATCG
>EAS20_8_6_1_6_1272/1
TACCTTCGCACTGCCCCGAGCTCGAACAGCTTGGCGCACCAACGGTATTGCCGGAATTACTCAAGAGCGA
>EAS20_8_6_1_6_1272/2
AGATATTCAACAGGATCTTCCAGCGTCAGAATATGCGCATCGGCATGTTGATTGAGATAGCCAACCATCG
>EAS20_8_6_1_6_668/1
CGAAGTCGGGCGAAATGGCGTGATTTCAGGGAGGATTTTCCAGAACATCAACGCCGAGGCCAGCGCGAAA
```

which is a bunch of sequence reads in something called the FASTA format: a ‘>’ character followed by a name (in this case computer generated, and more or less random), with a sequence right after it.

These sequences are generated by an instrument that takes shredded DNA and “digitizes” it – in this case it’s an Illumina sequencer, but there are many other such machines.

4.4 Assembling the data

Now run the following two commands:

```
velveth g.31 31 -shortPaired ecoli-reads-5m-dn-paired.fa.gz
velvetg g.31 -exp_cov auto
```

The first command tells the Velvet assembler to load the sequences into the directory ‘g.31’, using the value ‘31’ for the required ‘k’ parameter value. (More on ‘k’ later...) The name ‘g.31’ is just our way of keeping track of things – this can be any filename. I’m using ‘g’ for ‘genome’ and ‘31’ to remind me of what ‘k’ value I used.

The second command tells Velvet to assemble the shorter sequences into longer contiguous sequences, or “contigs”.

This essentially does what we did manually in class: looks for overlaps, and then sticks the sequences together.

The output of the last command should end with something like:

```
Final graph has 3590 nodes and n50 of 2328, max 11865, total 4580179, using 366068/371922 reads
```

which tells you a few statistics about the assembly –

- Velvet could assemble about 4000 sequences (3590 “nodes”);
- the N50 of the sequences was about 2.3 kb (2328 bases) – more on N50 later;
- the maximum contig length assembled is 11kb, which means hundreds of those little reads were put together;
- the sum of the bases that were assembled is about 4.5 mb (4580179), which is pretty close to the size of the E. coli genome!

The output of all of this is in the file ‘g.31/contigs.fa’, which you can look at using ‘head’ again:

```
head g.31/contigs.fa
```

You should see something like

```
>NODE_1_length_1698_cov_3.221437      CCTGTTTATCTTGCCCGGCCCATAGGCAATCT-
GTAACCAGTCAGCAATTTGGTTATTGC          TGAGTGCTGATTTTAGTGCAAACCATGA-
CAAAGCTGGCTGAGTATTACCTTGCGAGCTT      CAATAATCAATGCATCATAGGCGTTAT-
TAACAGCACTCTTCGCCGCGGGACTCGCTGCTA    AAAATGCGGCAGTAAGAAGTTTCAAAGC-
CCATTTGGTTTTTCGGGCACCTTTTTCTGCTAC    TTGAATACATCCTGTATTACTCCATGTATTGC-
CAAAATCTCTCTGTATCTAATTACAG            GTAAGTGAAGAAAGATATTTTGCACCT-
```



```

CATAATCCGTTATTAAACGCGGAAGAGAGA   CGTGAATTGTTGATGATGAGAAGAAGAAAT-
GATGAGCAGAGTGTCCATATAAAATCCTTT   TCTCGCCCGAAAATCCATTCCAATGATGAG-
GATCTTCAGGAATACGGCATAAATCCCAAT   GCCTTTTTCAAAATAAATTAGGATTAATAAT-
TAAATCAGTAAATTCCGATGCATGATT

```

4.5 Running multiple assemblies

Do one more assembly – for example, set the ‘k’ parameter to 21 (you can set it to any odd number between 19 and 51, if you want to try something different than 21).

```

velveth g.21 21 -shortPaired ecoli-reads-5m-dn-paired.fa.gz
velvetg g.21 -exp_cov auto

```

Now we have *two* assemblies... the second one should look like this:

Final graph has 2060 nodes and n50 of 6284, max 36734, total 4526331, using 370625/371922 reads

Is this better or worse than the k=31 assembly? Why?

Generate a few more assemblies – work with a pal to cover more ground. You should keep track of the velvetg statistics output; if you lose it, you can recover it by doing ‘tail g.31/Log’.

You can try:

- varying k by choosing any odd number between 19 and 51;
- removing the ‘-exp_cov auto’ command from ‘velvetg’;
- adding ‘-scaffolding no’ to the ‘velvetg’ command;
- Adding more read data. Grab this file:

<https://s3.amazonaws.com/public.ged.msu.edu/ecoli-reads-5m-dn-orphan.fa.gz>

using ‘curl’ as above, and then append ‘-short ecoli-reads-5m-dn-orphan.fa.gz’ to the ‘velveth’ command line. (The ‘velvetg’ command doesn’t need to change.)

Which of these assemblies is “best” by some criterion? Can you find an assembly that is “best” by more than one (unrelated) criterion?

4.6 Finishing up for today

Just leave your EC2 instances running so that we can access the data tomorrow.

Tomorrow, we’ll cover ways of graphing some of your statistics. One possible project to present on Friday is your analysis of these various assemblies.

Update: [Here’s an IPython Notebook](#) that shows you average length stats for a bunch of assemblies.

4.7 Questions and thoughts to address

Things to meditate upon –

- how do we manage complexity? Do we need to understand all these commands? What does each command do? In detail?

- why don't we have a nice user interface? Why is everything typing!?
- why are we using this Amazon machine rather than the computer in front of us?
- what is source code, anyway?
- why do the assemblies change when you change k?
- why might you get different numbers than me out of the velvet commands, sometime? The data going in isn't changing...?
- combinatorial explosion of parameters!!!

4.8 Reading

Genome sequence assembly primer

What does the k parameter do in assembly?

Assembly algorithms for next-generation sequencing data

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*