

---

# **labibi Documentation**

***Release 1.0***

**C. Titus Brown**

October 13, 2016



<b>1</b>	<b>Welcome!</b>	<b>3</b>
1.1	1. Learning goals . . . . .	3
1.2	2. Safe space and code of conduct . . . . .	3
1.3	3. Instructor introductions . . . . .	3
1.4	4. Amazon and cloud computing - why?! . . . . .	3
1.5	5. Sticky notes and how they work... + Minute Cards . . . . .	4
<b>2</b>	<b>Starting up an Amazon Web Services machine</b>	<b>5</b>
2.1	Start here: <i>Start an Amazon Web Services computer</i> : . . . . .	5
2.2	Full table of contents: . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>33</b>
<b>4</b>	<b>Short read quality and trimming</b>	<b>35</b>
4.1	Prepping the computer . . . . .	35
4.2	Installing some software . . . . .	35
4.3	Running Jupyter Notebook . . . . .	36
4.4	Data source . . . . .	36
4.5	1. Copying in some data to work with. . . . .	36
4.6	1. Copying data into a working location . . . . .	37
4.7	2. FastQC . . . . .	37
4.8	3. Trimmomatic . . . . .	38
4.9	4. FastQC again . . . . .	39
<b>5</b>	<b>K-mer Spectral Error Trimming</b>	<b>41</b>
5.1	Why (or why not) do k-mer trimming? . . . . .	43
<b>6</b>	<b>Run the MEGAHIT assembler</b>	<b>45</b>
6.1	While the assembly runs... . . . .	45
6.2	After the assembly is finished . . . . .	47
<b>7</b>	<b>Annotation with Prokka</b>	<b>49</b>
7.1	Installing Prokka . . . . .	49
7.2	Running Prokka . . . . .	49
7.3	References . . . . .	50
<b>8</b>	<b>Gene Abundance Estimation with Salmon</b>	<b>51</b>
8.1	Installing Salmon . . . . .	51
8.2	Running Salmon . . . . .	51

8.3	Working with count data . . . . .	52
8.4	Plotting the results . . . . .	53
8.5	References . . . . .	53
<b>9</b>	<b>Day 2 - installation instructions</b>	<b>55</b>
9.1	Running Jupyter Notebook . . . . .	55
<b>10</b>	<b>Technical information</b>	<b>57</b>

This workshop was given on October 12th and 13th, 2016, by Harriet Alexander and C. Titus Brown, at the Scripps Institute of Oceanography.

For more information, please [contact Titus directly](#).

We have an [EtherPad](#) for sharing text and asking questions.

Rough schedule:

- Wed, 9am-noon: Amazon Web Services; short reads, QC, and trimming.
- Logging into AWS
- Read quality evaluation and FASTQ trimming (FASTQC, Trimmomatic)
- (Optional) K-mer spectral error analysis and trimming (khmer)
- Wed, 1-4pm: Assembly, annotation, and evaluation.
- Short-read assembly with MEGAHIT
- Advantages and drawbacks to assembly (discussion)
- Evaluating assembly results (mapping rates, mapping viz, etc.)
- Annotating assembly (Prokka)
- Abundance calculations
- Thursday: topics TBD from below, or others:
  - Biology and bioinformatics:
  - Abundance comparisons between samples
  - Taxonomic analysis of reads and assemblies
  - Genome binning
  - CIRCOS plots
  - ShotMap for annotating shotgun reads
  - Moar computing:
  - Jupyter Notebook or RMarkdown
  - git for version control
  - Docker execution environments
  - Workflow strategies (scripting, make, doit, etc.)
  - More Amazon Web Services: EC2, S3, ...?
  - Miscellany:
  - Visualizing assembly graphs
  - MinHash sketches for comparing genomes and metagenomes
  - Tricks with sequences using Python libraries (screed & khmer)

Tutorials:



---

## Welcome!

---

### 1.1 1. Learning goals

For you:

- get a first (or second) look at tools;
- gain some experience in the basic command line;
- get 80% of way to a complete analysis of some data;
- introduction to philosophy and perspective of data analysis in science;

### 1.2 2. Safe space and code of conduct

This is intended to be a safe and friendly place for learning!

Please see the Software Carpentry workshop Code of Conduct: <http://software-carpentry.org/conduct.html>

In particular, please ask questions, because I guarantee you that your question will help others!

### 1.3 3. Instructor introductions

Harriet Alexander - postdoc at UC Davis.

Titus Brown - prof at UC Davis in the School of Vet Med.

### 1.4 4. Amazon and cloud computing - why?!

- simplifies software installation;
- can be used for bigger analyses quite easily;
- good for “burst” capacity (just got a data set!)
- accessible everywhere;

## 1.5 5. Sticky notes and how they work... + Minute Cards

Basic rules:

- no sticky note - “working on it”
- green sticky note - “all is well”
- red sticky note - “need help!”

Place the sticky notes where we can see them from the back of the room – e.g. on the back of your laptop.

At the end of each session (coffee break, lunch, end of day) please write down on an index card **one thing you learned** and **one thing you’re still confused about**.

—

Next: `n-overview`



---

## Starting up an Amazon Web Services machine

---

### 2.1 Start here: *Start an Amazon Web Services computer:*

### 2.2 Full table of contents:

#### 2.2.1 Start an Amazon Web Services computer:

This page shows you how to create a new “AWS instance”, or a running computer.

---

Start at the Amazon Web Services console (<http://aws.amazon.com/> and sign in to the console).



## 0. Select “EC2 - virtual servers in the cloud”

### Amazon Web Services

#### Compute

- EC2**  
Virtual Servers in the Cloud
- EC2 Container Service  
Run and Manage Docker Containers
- Elastic Beanstalk  
Run and Manage Web Apps
- Lambda  
Run Code in Response to Events

#### Storage & Content Delivery

- S3  
Scalable Storage in the Cloud
- CloudFront  
Global Content Delivery Network
- Elastic File System **PREVIEW**  
Fully Managed File System for EC2
- Glacier  
Archive Storage in the Cloud
- Import/Export Snowball  
Large Scale Data Transport
- Storage Gateway  
Hybrid Storage Integration

#### Database

- RDS  
Managed Relational Database Service
- DynamoDB  
Managed NoSQL Database
- ElastiCache  
In-Memory Cache
- Redshift  
Fast, Simple, Cost-Effective Data Warehousing

#### Developer Tools

- CodeCommit  
Store Code in Private Git Repositories
- CodeDeploy  
Automate Code Deployments
- CodePipeline  
Release Software using Continuous Delivery

#### Management Tools

- CloudWatch  
Monitor Resources and Applications
- CloudFormation  
Create and Manage Resources with Templates
- CloudTrail  
Track User Activity and API Usage
- Config  
Track Resource Inventory and Changes
- OpsWorks  
Automate Operations with Chef
- Service Catalog  
Create and Use Standardized Products
- Trusted Advisor  
Optimize Performance and Security

#### Security & Identity

- Identity & Access Management  
Manage User Access and Encryption Keys
- Directory Service  
Host and Manage Active Directory
- Inspector **PREVIEW**  
Analyze Application Security
- WAF  
Filter Malicious Web Traffic

#### Internet of Things

- AWS IoT  
Connect Devices to the Cloud

#### Game Development

- GameLift  
Deploy and Scale Session-based Multiplayer Games

#### Mobile Services

- Mobile Hub  
Build, Test, and Monitor Mobile Apps
- Cognito  
User Identity and App Data Synchronization
- Device Farm  
Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
- Mobile Analytics  
Collect, View and Export App Analytics
- SNS  
Push Notification Service

#### Application Services

- API Gateway  
Build, Deploy and Manage APIs
- AppStream  
Low Latency Application Streaming
- CloudSearch  
Managed Search Service
- Elastic Transcoder  
Easy-to-Use Scalable Media Transcoding
- SES  
Email Sending and Receiving Service
- SQS

### Resource Groups

[Learn more](#)

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

[Create a Group](#)
[Tag Editor](#)

### Additional Resources

#### Getting Started

Read our [documentation](#) or view our [training](#) to learn more about AWS.

#### AWS Console Mobile App

View your resources on the go with our AWS Console mobile app, available from [Amazon Appstore](#), [Google Play](#), or [iTunes](#).

#### AWS Marketplace

Find and buy software, launch with 1-Click and pay by the hour.

[AWS re:Invent Announcements](#)  
Explore the next generation of AWS cloud capabilities. [See what's new](#)

### Service Health

✓ All services operating normally.

## 1. Switch to zone US West (N California)

The screenshot shows the AWS Management Console interface for the N. California region. The left sidebar contains navigation links for EC2 Dashboard, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area is divided into several sections: Resources (listing various EC2 resources), Create Instance (with a highlighted 'Launch Instance' button), Service Health (showing 'US West (N. California): This service is operating normally'), and Scheduled Events. The top navigation bar shows the user is logged in as Titus Brown in the N. California region.

## 2. Click on “Launch instance.”

## 3. Select “Community AMIs.”

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' page in the AWS console. The 'Quick Start' section is active, and the 'Community AMIs' tab is selected. The 'Amazon Linux AMI 2015.09.2 (HVM), SSD Volume Type - ami-d1f482b1' is highlighted. The 'Select' button is visible. The page also includes a 'Cancel and Exit' link and a progress bar at the top.

## 2.2. Full table of contents:

AWS Marketplace

Community AMIs

Amazon Linux

Free tier eligible

Amazon Linux AMI 2015.09.2 (HVM), SSD Volume Type - ami-d1f482b1

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs    Virtualization type: hvm

Select

64-bit

AWS

Services

Edit

Titus Brown

N. California

Support

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Tag Instance

6. Configure Security Group

7. Review

Cancel and Exit

### Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Operating system

☐ Amazon Linux
☐ Cent OS
☐ Debian
☐ Fedora
☐ Gentoo
☐ OpenSUSE
☐ Other Linux
☐ Red Hat

ubuntu/images/hvm/ubuntu-wily-15.10-amd64-server-20160222 - ami-05384865

Root device type: ebs    Virtualization type: hvm

Select

64-bit

< > 1 to 1 of 1 AMIs > >

Feedback

English

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

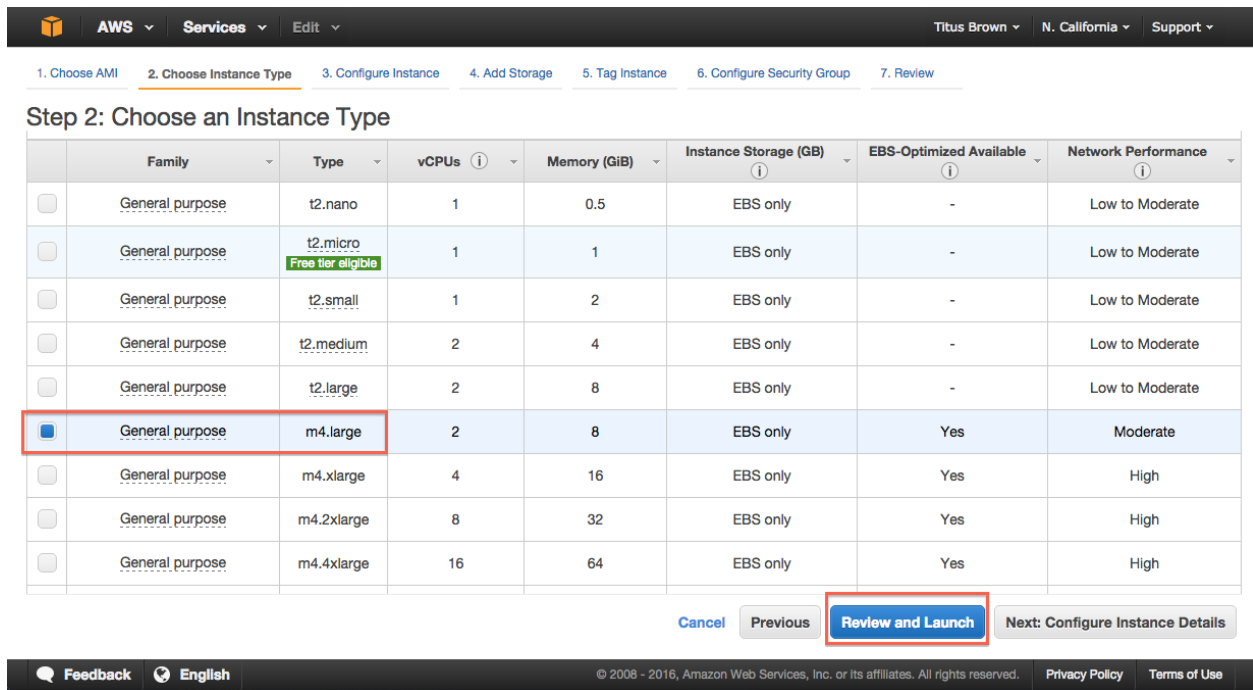
Privacy Policy

Terms of Use



## 5. Click on “Select.”

## 6. Choose m4.xlarge.



Step 2: Choose an Instance Type

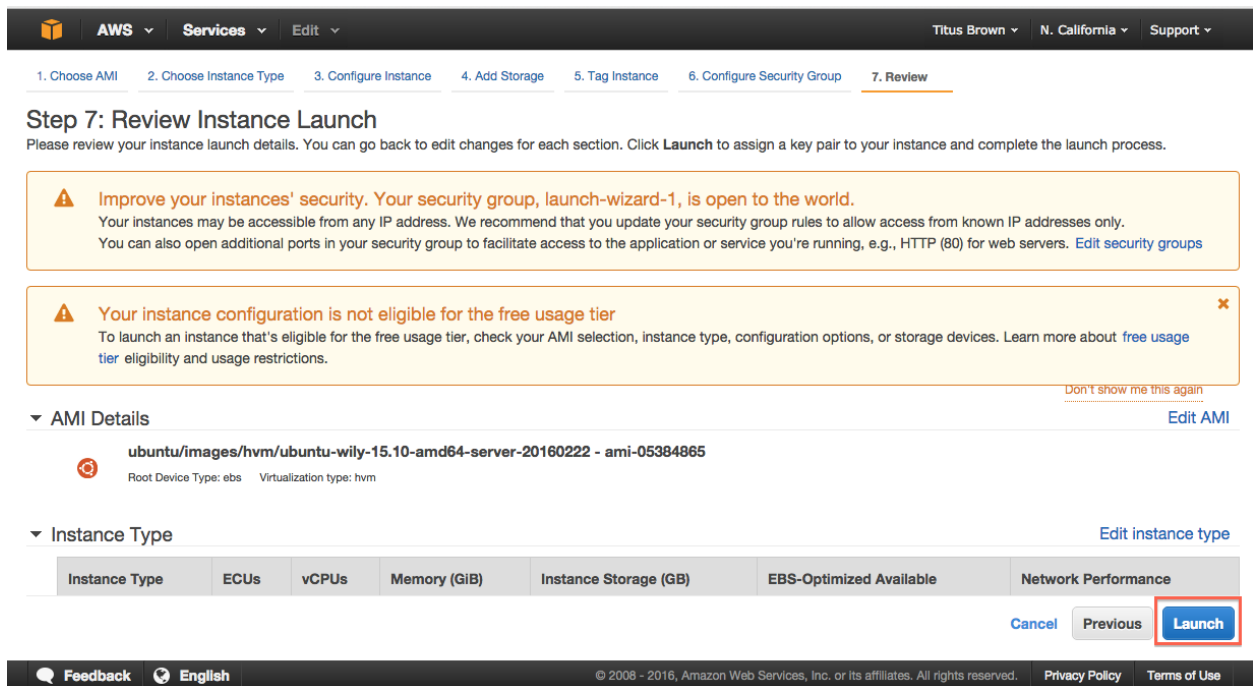
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate
<input type="checkbox"/>	General purpose	m4.xlarge	4	16	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.4xlarge	16	64	EBS only	Yes	High

Cancel Previous **Review and Launch** Next: Configure Instance Details

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## 7. Click “Review and Launch.”

## 8. Click “Launch.”



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**⚠ Improve your instances' security. Your security group, launch-wizard-1, is open to the world.**  
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

**⚠ Your instance configuration is not eligible for the free usage tier**  
To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about [free usage tier eligibility and usage restrictions](#).

Don't show me this again

▼ AMI Details [Edit AMI](#)

ubuntu/images/hvm/ubuntu-wily-15.10-amd64-server-20160222 - ami-05384865  
Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
m4.xlarge	4	4	16	16	Yes	High

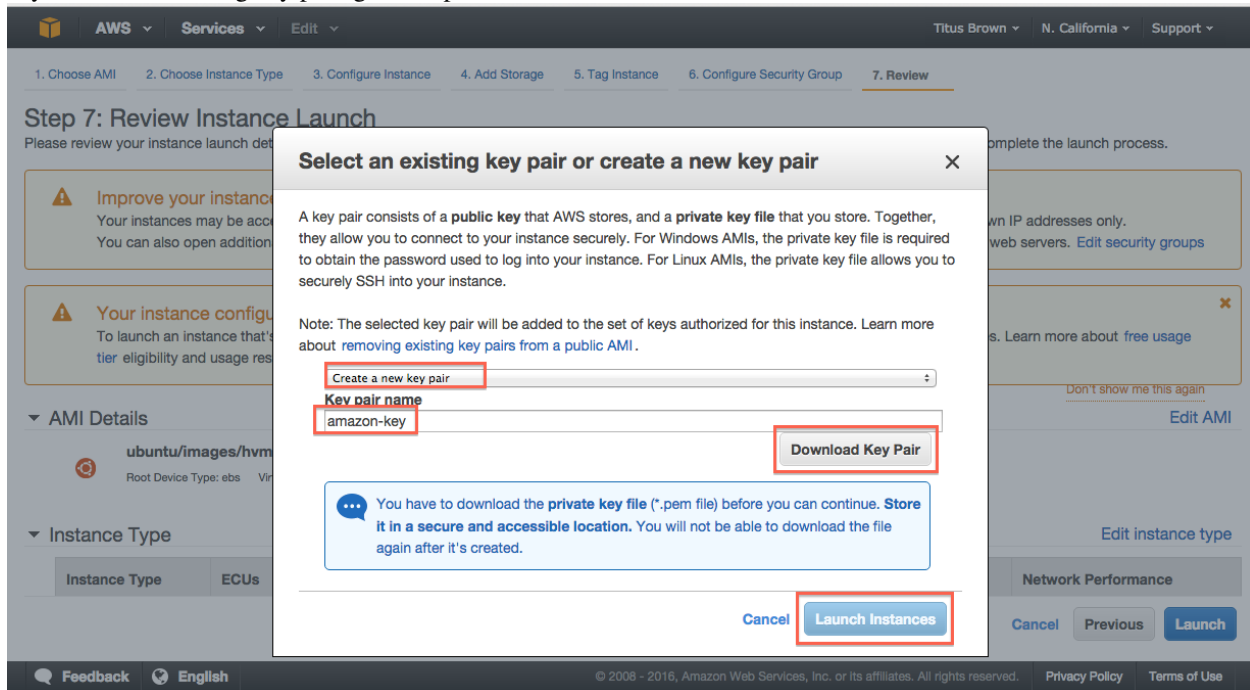
Cancel Previous **Launch**

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## 9. Select “Create a new key pair.”

Note: you only need to do this the first time you create an instance. If you know where your amazon-key.pem file is, you can select “Use an existing key pair” here. But you can always create a new key pair if you want, too.

If you have an existing key pair, go to step 12, “Launch instance.”

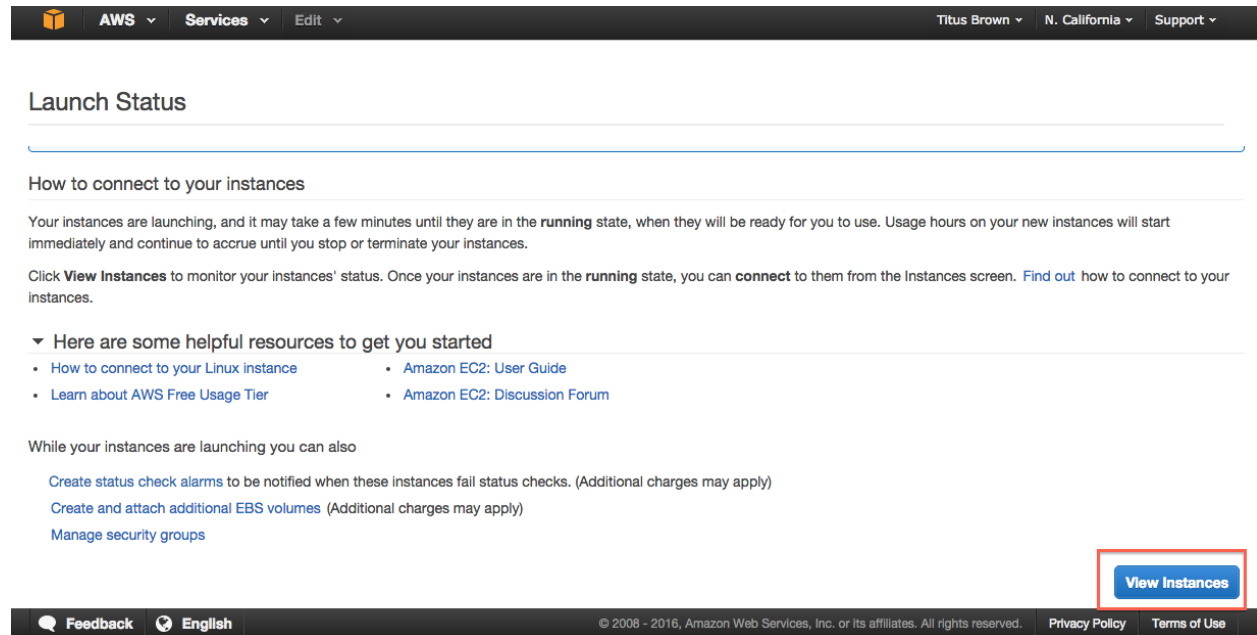


10. Enter name ‘amazon-key’.

11. Click “Download key pair.”

12. Click “Launch instance.”

13. Select View instances (lower right)



**Launch Status**

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to your instances.

▼ Here are some helpful resources to get you started

- [How to connect to your Linux instance](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

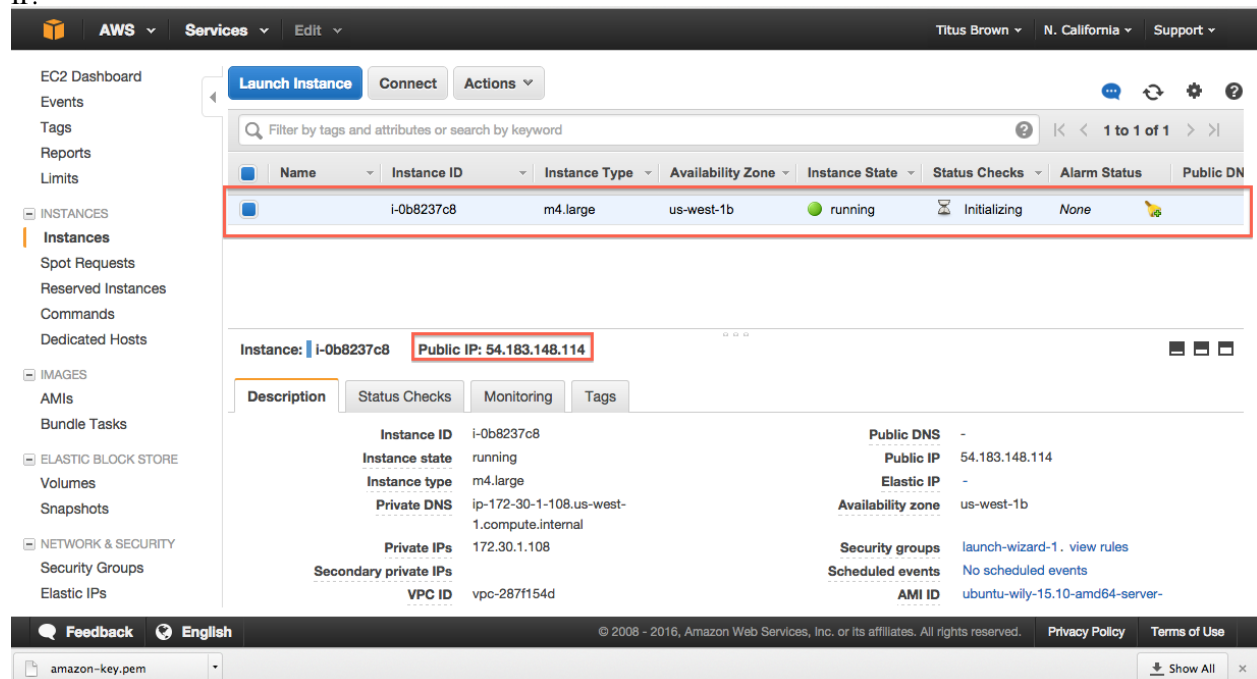
While your instances are launching you can also

- [Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)
- [Create and attach additional EBS volumes](#) (Additional charges may apply)
- [Manage security groups](#)

**View Instances**

14. Bask in the glory of your running instance

Note that for your instance name you can use either “Public IP” or “Public DNS”. Here, the machine only has a public IP.



**EC2 Dashboard**

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-0b8237c8	m4.large	us-west-1b	running	Initializing	None	

Instance: i-0b8237c8 **Public IP: 54.183.148.114**

**Description** Status Checks Monitoring Tags

Property	Value	Property	Value
Instance ID	i-0b8237c8	Public DNS	-
Instance state	running	Public IP	54.183.148.114
Instance type	m4.large	Elastic IP	-
Private DNS	ip-172-30-1-108.us-west-1.compute.internal	Availability zone	us-west-1b
Private IPs	172.30.1.108	Security groups	launch-wizard-1. view rules
Secondary private IPs	-	Scheduled events	No scheduled events
VPC ID	vpc-287f154d	AMI ID	ubuntu-wily-15.10-amd64-server-



You can now [Log into your instance with the UNIX shell](#) or [Configure your instance firewall](#).

## 2.2.2 Log into your instance with the UNIX shell

You will need the `amazon-key.pem` file that was downloaded in step 11 of booting up your new instance (see [Start an Amazon Web Services computer](#):).

Then, you can either [Log into your instance from a Mac or Linux machine](#) or [Log into your instance from a Windows machine](#).

### Log into your instance via the UNIX shell (Mac/Linux)

See: [Log into your instance from a Mac or Linux machine](#)

### Log into your instance via MobaXTerm (Windows)

See: [Log into your instance from a Windows machine](#)

---

Logging in is the starting point for most of the follow-on tutorials. For example, you can now install and run software on your EC2 instance.

Go back to the top page to continue: [Starting up an Amazon Web Services machine](#)

## 2.2.3 Log into your instance from a Mac or Linux machine

You'll need to do two things: first, set the permissions on `amazon-key.pem`:

```
chmod og-rwx ~/Downloads/amazon-key.pem
```

Then, ssh into your new machine using your key:

```
ssh -i ~/Downloads/amazon-key.pem -l ubuntu MACHINE_NAME
```

where you should replace `MACHINE_NAME` with the public IP or hostname of your EC2 instance, which is located at the top of the host information box (see screenshot below). It should be something like `54.183.148.114` or `ec2-XXX-YYY.amazonaws.com`.

Here are some screenshots!



## Change permissions and execute ssh

```
% chmod og-rwx ~/Downloads/amazon-key.pem
% ssh -i ~/Downloads/amazon-key.pem ubuntu@54.183.148.114
The authenticity of host '54.183.148.114 (54.183.148.114)' can't be established.
RSA key fingerprint is b6:de:2f:fb:e7:12:e5:1e:5d:66:37:ef:40:bb:b7:c8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.183.148.114' (RSA) to the list of known hosts.
```

## Successful login

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.183.148.114' (RSA) to the list of known hosts.
Welcome to Ubuntu 15.10 (GNU/Linux 4.2.0-30-generic x86_64)

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

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

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-30-1-108:~$
```

## Host information box - MACHINE\_NAME location

The screenshot shows the AWS Management Console interface. On the left, there is a navigation menu with options like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area displays details for an EC2 instance named 'i-0b8237c8'. The instance is in a 'running' state with a 'Public IP' of 54.183.148.114. The console also shows a table of instance details with columns for Instance ID, Instance state, Instance type, Private DNS, Private IPs, Secondary private IPs, VPC ID, Public DNS, Public IP, Elastic IP, Availability zone, Security groups, Scheduled events, and AMI ID.

Instance ID	Instance state	Instance type	Private DNS	Private IPs	Secondary private IPs	VPC ID	Public DNS	Public IP	Elastic IP	Availability zone	Security groups	Scheduled events	AMI ID
i-0b8237c8	running	m4.large	ip-172-30-1-108.us-west-1.compute.internal	172.30.1.108		vpc-287f154d	54.183.148.114			us-west-1b	launch-wizard-1. view rules	No scheduled events	ubuntu-wily-15.10-amd64-server-

Logging in is the starting point for most of the follow-on tutorials. For example, you can now install and run software on your EC2 instance.

Go back to the top page to continue: [Starting up an Amazon Web Services machine](#)

## 2.2.4 Log into your instance from a Windows machine

Go follow the instructions this URL:

<https://angus.readthedocs.org/en/2015/amazon/log-in-with-mobaxterm-win.html>

Logging in is the starting point for most of the follow-on tutorials. For example, you can now install and run software on your EC2 instance.

Go back to the top page to continue: [Starting up an Amazon Web Services machine](#)

## 2.2.5 Configure your instance firewall

Normally, Amazon computers only allow shell logins via ssh (port 22 access). If we want to run a Web service or something else, we need to give the outside world access to other network locations on the computer.

Below, we will open ports 8000-9000, which will let us run things like RStudio Server. If you want to run other things, like a Web server, you'll need to find the port(s) associated with those services and open those instead of 8000-9000. (Tip: Web servers run on port 80.)

### 1. Select 'Security Groups'

Find "Security Groups" in the lower pane of your instance's information page, and click on "launch-wizard-N".

The screenshot shows the AWS Management Console interface. On the left, there's a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The 'Security Groups' link under NETWORK & SECURITY is highlighted. The main panel displays the details for instance 'i-0b8237c8'. The 'Security groups' field is highlighted with a red box, showing 'launch-wizard-1' with a link to 'view rules'.

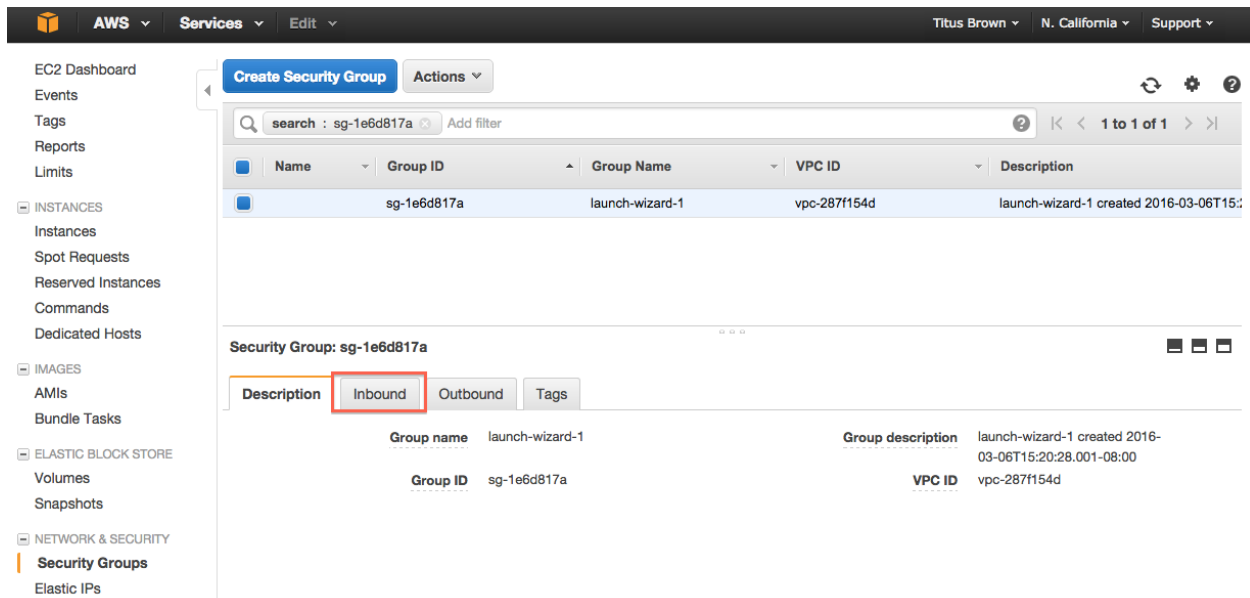
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-0b8237c8	m4.large	us-west-1b	running	Initializing	None	

Instance: **i-0b8237c8** Public IP: 54.183.148.114

Field	Value
Instance ID	i-0b8237c8
Instance state	running
Instance type	m4.large
Private DNS	ip-172-30-1-108.us-west-1.compute.internal
Private IPs	172.30.1.108
Secondary private IPs	
VPC ID	vpc-2871f54d
Public DNS	-
Public IP	54.183.148.114
Elastic IP	-
Availability zone	us-west-1b
Security groups	launch-wizard-1 <a href="#">view rules</a>
Scheduled events	No scheduled events
AMI ID	ubuntu-wily-15.10-amd64-server-



## 2. Select 'Inbound'



EC2 Dashboard  
Events  
Tags  
Reports  
Limits

INSTANCES  
Instances  
Spot Requests  
Reserved Instances  
Commands  
Dedicated Hosts

IMAGES  
AMIs  
Bundle Tasks

ELASTIC BLOCK STORE  
Volumes  
Snapshots

NETWORK & SECURITY  
**Security Groups**  
Elastic IPs

**Create Security Group** Actions

search : sg-1e6d817a Add filter

Name	Group ID	Group Name	VPC ID	Description
	sg-1e6d817a	launch-wizard-1	vpc-287f154d	launch-wizard-1 created 2016-03-06T15:00:00.000Z

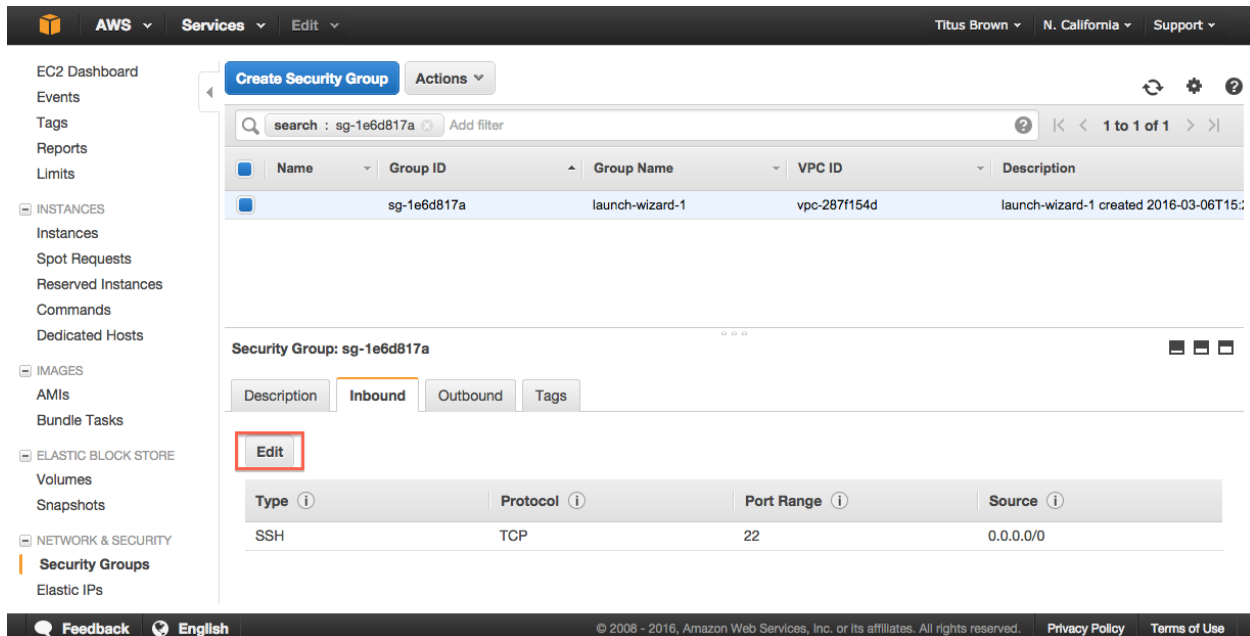
Security Group: sg-1e6d817a

Description Inbound Outbound Tags

Group name launch-wizard-1 Group description launch-wizard-1 created 2016-03-06T15:20:28.001-08:00

Group ID sg-1e6d817a VPC ID vpc-287f154d

## 3. Select 'Edit'



EC2 Dashboard  
Events  
Tags  
Reports  
Limits

INSTANCES  
Instances  
Spot Requests  
Reserved Instances  
Commands  
Dedicated Hosts

IMAGES  
AMIs  
Bundle Tasks

ELASTIC BLOCK STORE  
Volumes  
Snapshots

NETWORK & SECURITY  
**Security Groups**  
Elastic IPs

**Create Security Group** Actions

search : sg-1e6d817a Add filter

Name	Group ID	Group Name	VPC ID	Description
	sg-1e6d817a	launch-wizard-1	vpc-287f154d	launch-wizard-1 created 2016-03-06T15:00:00.000Z

Security Group: sg-1e6d817a

Description Inbound Outbound Tags

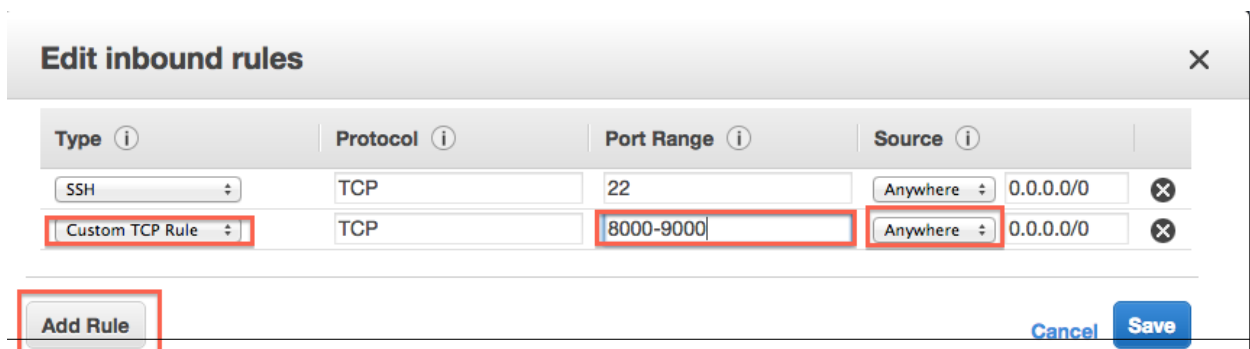
Edit

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0

Feedback English

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## 4. Select 'Add Rule'



**Edit inbound rules**

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0
Custom TCP Rule	TCP	8000-9000	Anywhere 0.0.0.0/0

Add Rule Cancel Save

## 5. Enter rule information

Add a new rule: Custom TCP 8000-9000. Source Anywhere

6. Select 'Save'.

7. Return to the Instances page.

The screenshot displays the AWS Management Console interface for creating a security group. The left-hand navigation pane shows the 'Instances' link highlighted with a red rectangle. The main content area is titled 'Create Security Group' and features a search bar with the text 'sg-1e6d817a'. Below the search bar is a table listing security groups. The table has columns for Name, Group ID, Group Name, VPC ID, and Description. A single entry is shown: 'sg-1e6d817a', 'launch-wizard-1', 'vpc-287f154d', and 'launch-wizard-1 created 2016-03-06T15:'. Below the table, the 'Security Group: sg-1e6d817a' section is visible, with tabs for Description, Inbound, Outbound, and Tags. The 'Inbound' tab is selected, showing an 'Edit' button and a table of inbound rules. The table has columns for Type, Protocol, Port Range, and Source. Two rules are listed: 'SSH' (TCP, Port Range 22, Source 0.0.0.0/0) and 'Custom TCP Rule' (TCP, Port Range 8000 - 9000, Source 0.0.0.0/0).

You're done!

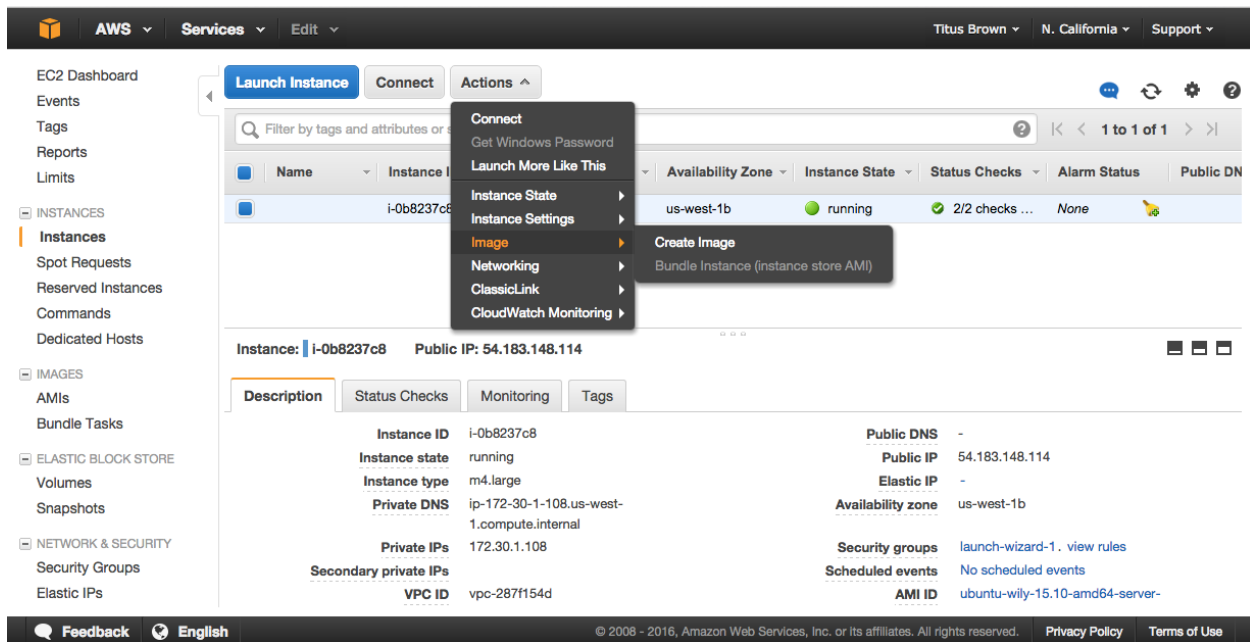
Go back to the index: [Starting up an Amazon Web Services machine](#)





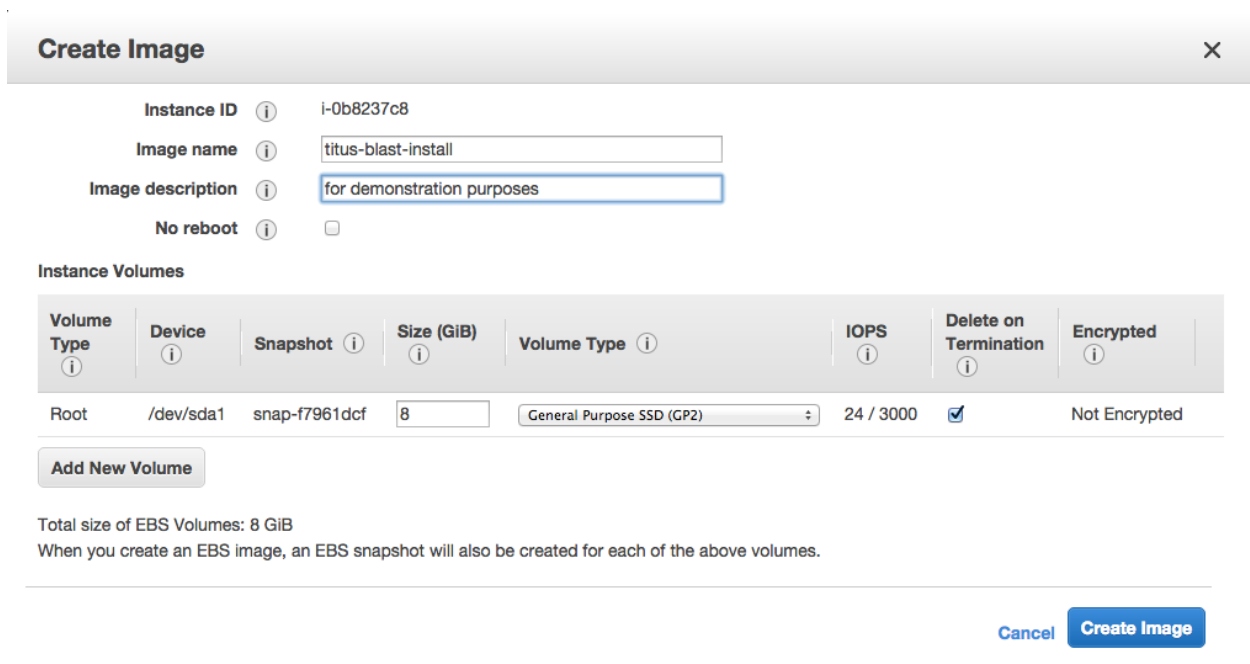
## 2.2.6 Creating your own Amazon Machine Image

### 1. Actions, Create image



The screenshot shows the AWS Management Console interface. On the left, the navigation pane is visible with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area shows an EC2 instance named 'i-0b8237c8' in the 'us-west-1b' availability zone, with a state of 'running'. The 'Actions' menu is open, and 'Create Image' is highlighted. Below the instance details, the 'Description' tab is active, showing instance metadata such as Instance ID, Instance state, Instance type, Private DNS, Private IPs, Secondary private IPs, VPC ID, Public DNS, Public IP, Elastic IP, Availability zone, Security groups, Scheduled events, and AMI ID.

### 2. Fill out name and description



The 'Create Image' dialog box is shown. It contains the following fields and options:

- Instance ID:** i-0b8237c8
- Image name:** titus-blast-install
- Image description:** for demonstration purposes
- No reboot:** ☐

Below these fields is the **Instance Volumes** section, which displays a table of volumes:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/sda1	snap-f7961dcf	8	General Purpose SSD (GP2)	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted

Below the table, there is an 'Add New Volume' button. At the bottom of the dialog, there are 'Cancel' and 'Create Image' buttons.

### 3. Wait for it to become available

AWS

Services

Edit

Titus Brown

N. California

Support

EC2 Dashboard

Events

Tags

Reports

Limits

Launch

Actions

Owned by me

Filter by tags and attributes or search by keyword

1 to 1 of 1

<input type="checkbox"/>	Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
<input checked="" type="checkbox"/>	titus-blast-install	ami-2407c44	817232153141/tl...	817232153141	Private	pending	March 6, 2016 at 4:42	

## 2.2 Full table of contents:

Go back to the index: [Starting up an Amazon Web Services machine](#)

### 2.2.7 Working with persistent storage: volumes and snapshots

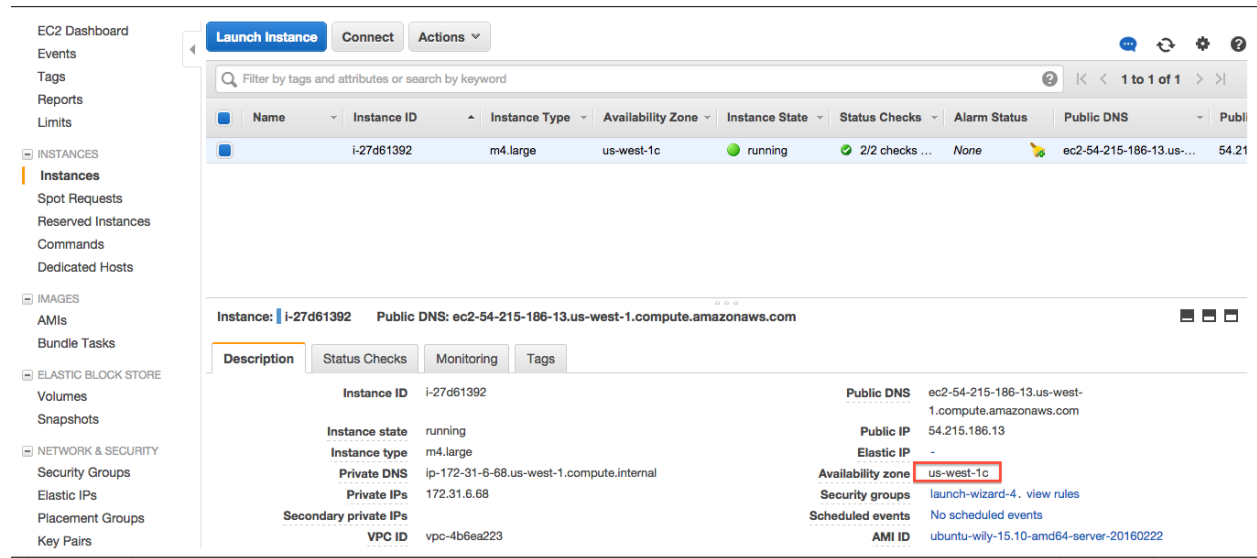
Volumes are basically UNIX disks (“block devices”) that will persist after you terminate your instance. They are tied to a zone within a region and can only be mounted on instances within that zone.

Snapshots are an Amazon-specific thing that let you communicate data on volumes between accounts. They are “read-only” backups that are created from volumes; they can be used to create new volumes in turn, and can also be shared with specific people (or made public). Snapshots are tied to a region but not a zone.



## Creating persistent volumes to store data

### 0. Locate your instance zone



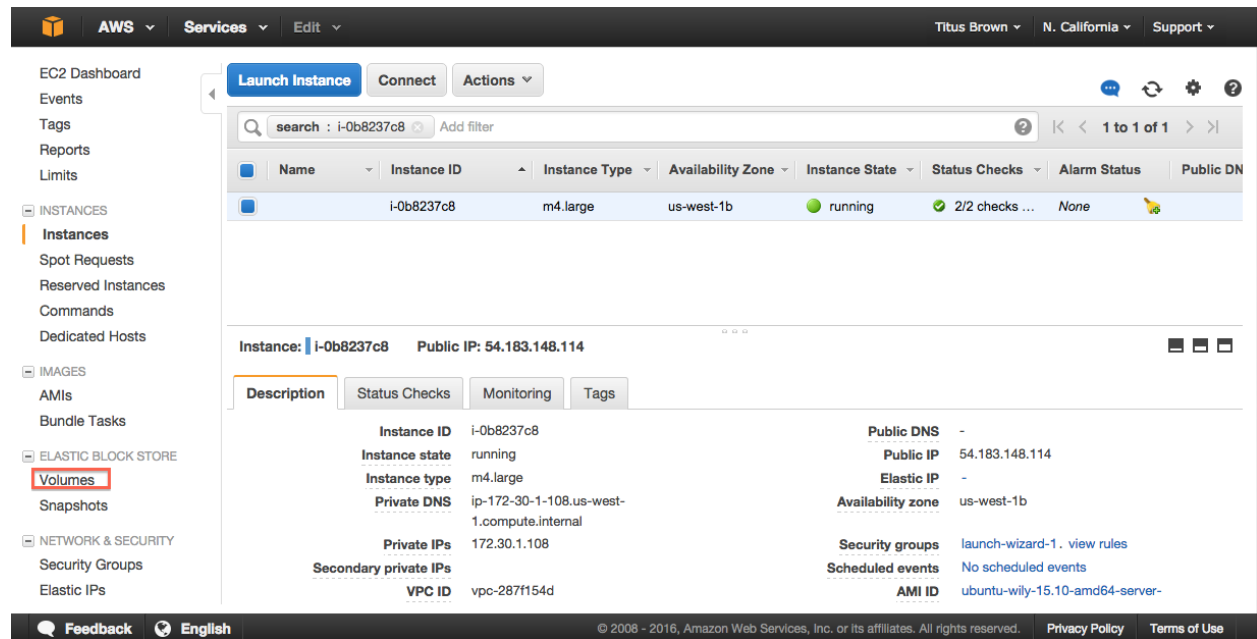
The screenshot shows the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area displays the details for instance i-27d61392. The instance is running in the us-west-1c availability zone. The 'Availability zone' field is highlighted with a red box.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
i-27d61392	i-27d61392	m4.large	us-west-1c	running	2/2 checks ...	None	ec2-54-215-186-13.us-west-1.compute.amazonaws.com	54.215.186.13

Instance: i-27d61392 Public DNS: ec2-54-215-186-13.us-west-1.compute.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID: i-27d61392	Instance state: running	Public DNS: ec2-54-215-186-13.us-west-1.compute.amazonaws.com	Public IP: 54.215.186.13
Instance type: m4.large	Private DNS: ip-172-31-6-68.us-west-1.compute.internal	Elastic IP: -	Availability zone: us-west-1c
Private IPs: 172.31.6.68	Secondary private IPs: -	Security groups: launch-wizard-4. view rules	Scheduled events: No scheduled events
VPC ID: vpc-4b6ea223		AMI ID: ubuntu-wily-15.10-amd64-server-20160222	

### 1. Click on the volumes tab



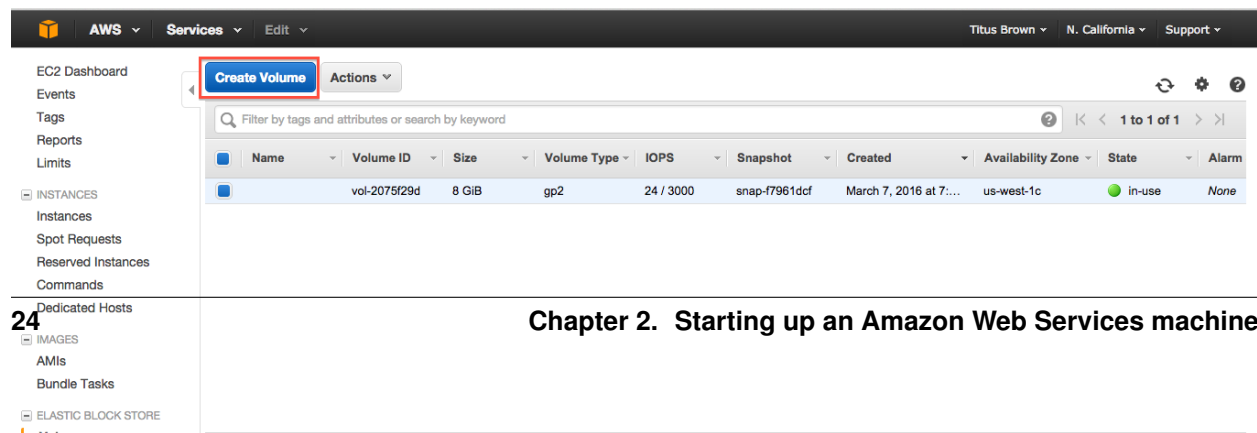
The screenshot shows the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The 'Volumes' tab is selected in the left sidebar. The main content area displays the details for instance i-0b8237c8. The instance is running in the us-west-1b availability zone. The 'Availability zone' field is highlighted with a red box.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
i-0b8237c8	i-0b8237c8	m4.large	us-west-1b	running	2/2 checks ...	None	54.183.148.114	54.183.148.114

Instance: i-0b8237c8 Public IP: 54.183.148.114

Description	Status Checks	Monitoring	Tags
Instance ID: i-0b8237c8	Instance state: running	Public DNS: -	Public IP: 54.183.148.114
Instance type: m4.large	Private DNS: ip-172-30-1-108.us-west-1.compute.internal	Elastic IP: -	Availability zone: us-west-1b
Private IPs: 172.30.1.108	Secondary private IPs: -	Security groups: launch-wizard-1. view rules	Scheduled events: No scheduled events
VPC ID: vpc-287f154d		AMI ID: ubuntu-wily-15.10-amd64-server-	

### 2. 'Create Volume'



The screenshot shows the AWS Management Console interface for the 'Create Volume' page. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The 'Create Volume' button is highlighted with a red box.

Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Zone	State	Alarm
vol-2075f29d	vol-2075f29d	8 GiB	gp2	24 / 3000	snap-f7961dcf	March 7, 2016 at 7:...	us-west-1c	in-use	None

Attach Volume

×

Volume ⓘ

vol-21e1a98e in us-west-1b

Instance ⓘ

in us-west-1b

Device ⓘ

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

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel

Attach

## 7. On your instance, list block devices

Type:

```
lsblk
```

You should see something like this:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
xvda	202:0	0	8G	0	disk	
└─xvda1	202:1	0	8G	0	part	/
xvdf	202:80	0	100G	0	disk	

Now format the disk (ONLY ON EMPTY DISKS - THIS WILL ERASE ANY DATA ON THE DISK):

```
sudo mkfs -t ext4 /dev/xvdf
```

and mount the disk:

```
sudo mkdir /disk
sudo mount /dev/xvdf /disk
sudo chmod a+rwxt /disk
```

and voila, anything you put on /disk will be on the volume that you allocated!

The command 'df -h' will show you what disks are actually mounted & where.

## Detaching volumes

### 1. Unmount it from the instance

Change out of the directory, stop any running programs using it, and then:

```
sudo umount /disk
```

### 2. Detach

On the 'volumes' tab in your EC2 console, go to Actions, Detach.

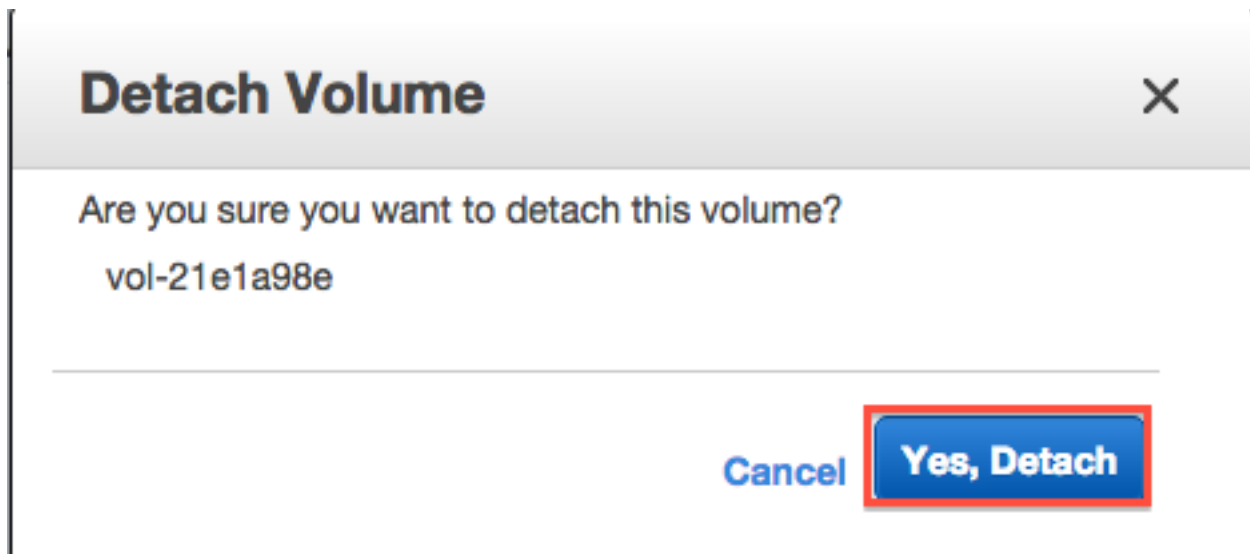
The screenshot shows the AWS Management Console interface. On the left, the navigation pane is visible with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The 'Volumes' link under 'ELASTIC BLOCK STORE' is selected. The main content area shows a list of volumes. A table lists two volumes: 'gp2' with IOPS '300 / 3000' and 'gp2' with IOPS '24 / 3000'. The first volume is selected. An 'Actions' dropdown menu is open, showing options: 'Delete Volume', 'Attach Volume', 'Detach Volume' (highlighted with a red box), 'Force Detach Volume', 'Create Snapshot', 'Change Auto-Enable IO Setting', and 'Add/Edit Tags'. Below the table, the details for volume 'vol-21e1a98e' are shown, including its ID, size (100 GiB), creation time, and availability zone.

Volume Type	IOPS	Snapshot	Created	Availability Zone
gp2	300 / 3000		March 6, 2016 at 4:...	us-west-1b
gp2	24 / 3000	snap-f7961dcf	March 6, 2016 at 3:...	us-west-1b

Volumes: **vol-21e1a98e**

Volume ID	Size	Created	Alarm status	Snapshot	Availability Zone
vol-21e1a98e	100 GiB	March 6, 2016 at 4:53:31 PM	None	-	us-west-1b

### 3. Yes, detach.

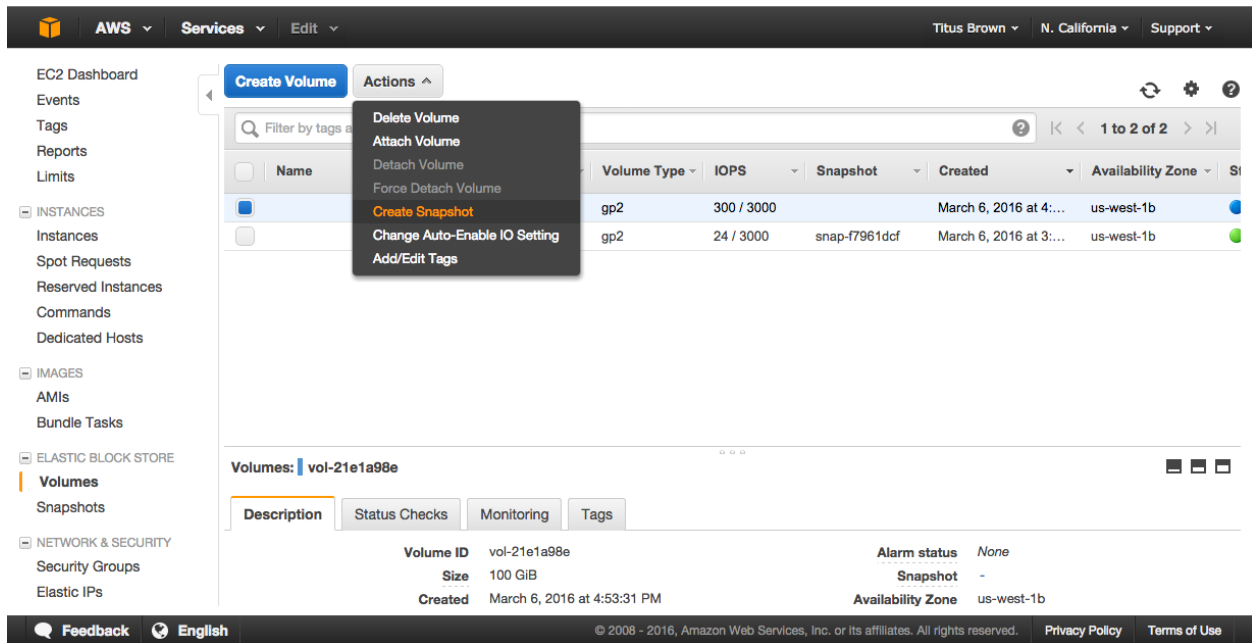


Note, volumes remain attached when you reboot or stop an instance, but are (of course) detached when you terminate an instance.



## Creating snapshots of volumes

### 1. Actions, Create snapshot



The screenshot shows the AWS Management Console interface. On the left, the navigation pane is visible with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The 'Volumes' link under 'ELASTIC BLOCK STORE' is selected. The main content area shows a list of volumes. A context menu is open over the first volume, 'vol-21e1a98e', with the 'Create Snapshot' option highlighted. Below the list, the details for 'vol-21e1a98e' are shown, including its size (100 GiB) and creation time (March 6, 2016 at 4:53:31 PM).

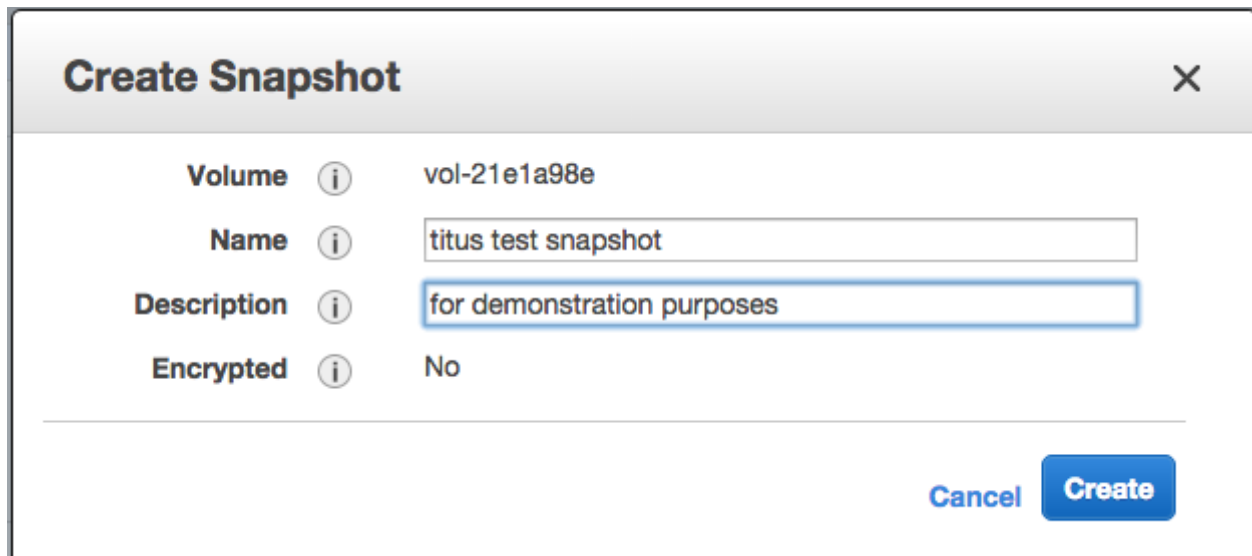
Volume Type	IOPS	Snapshot	Created	Availability Zone
gp2	300 / 3000		March 6, 2016 at 4:...	us-west-1b
gp2	24 / 3000	snap-f7961dcf	March 6, 2016 at 3:...	us-west-1b

Volumes: **vol-21e1a98e**

**Description** | Status Checks | Monitoring | Tags

Volume ID	vol-21e1a98e	Alarm status	None
Size	100 GiB	Snapshot	-
Created	March 6, 2016 at 4:53:31 PM	Availability Zone	us-west-1b

### 2. Fill out name and description

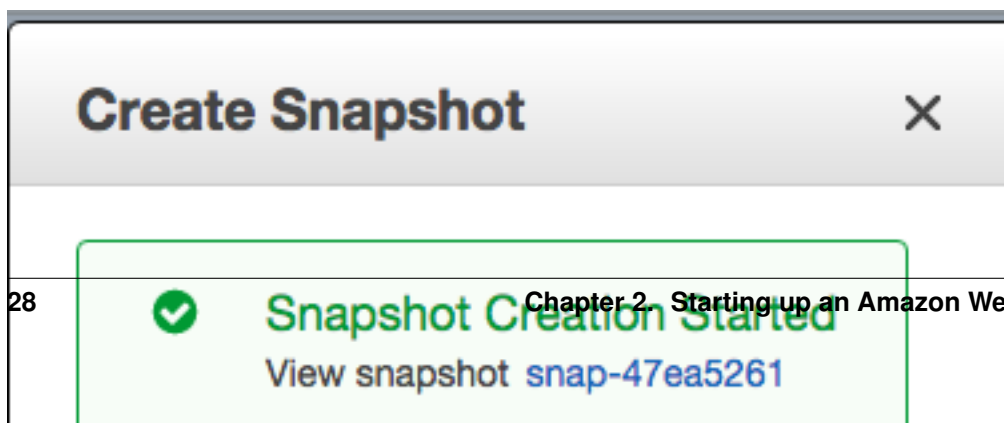


The 'Create Snapshot' dialog box is shown. It contains the following fields:

- Volume**: vol-21e1a98e
- Name**: titus test snapshot
- Description**: for demonstration purposes
- Encrypted**: No

At the bottom right, there are 'Cancel' and 'Create' buttons.

### 3. Click 'Close' & wait.



The 'Create Snapshot' dialog box is shown with a green checkmark and the message 'Snapshot Creation Started'. Below the message, it says 'View snapshot snap-47ea5261'.



## 2.2.8 Terminating your instance

Amazon will happily charge you for running instances and/or associated ephemeral storage until the cows come home - it's your responsibility to turn things off. The Right Way to do this for running instances is to terminate.

The caveat here is that *everything ephemeral* will be deleted (excluding volumes that you created/attached). So you want to make sure you transfer off anything you care about.

To terminate:

### 1. Select Actions, Instance State, Terminate

In the 'Instances' tab, select your instance and then go to the Actions menu.


The screenshot shows the AWS Management Console interface. On the left is a navigation sidebar with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main area displays the 'Instances' tab with a table of instances. One instance, 'i-0b8237c8', is selected. A context menu is open over this instance, showing options like 'Connect', 'Launch More Like This', 'Instance State', 'Instance Settings', 'Image', 'Networking', 'ClassicLink', and 'CloudWatch Monitoring'. The 'Instance State' sub-menu is expanded, showing 'Start', 'Stop', 'Reboot', and 'Terminate' (which is highlighted with a red box). Below the instance list, the details for instance 'i-0b8237c8' are shown, including its public IP (54.183.148.114) and various configuration details.

Instance: i-0b8237c8		Public IP: 54.183.148.114	
<div> <div>Description</div> <div>Status Checks</div> <div>Monitoring</div> <div>Tags</div> </div>			
Instance ID	i-0b8237c8	Public DNS	-
Instance state	running	Public IP	54.183.148.114
Instance type	m4.large	Elastic IP	-
Private DNS	ip-172-30-1-108.us-west-1.compute.internal	Availability zone	us-west-1b
Private IPs	172.30.1.108	Security groups	launch-wizard-1. <a href="#">view rules</a>
Secondary private IPs	-	Scheduled events	<a href="#">No scheduled events</a>
VPC ID	vpc-2b7f154d	AMI ID	ubuntu-wily-15.10-amd64-server-

## 2. Agree to terminate.

### Terminate Instances

×

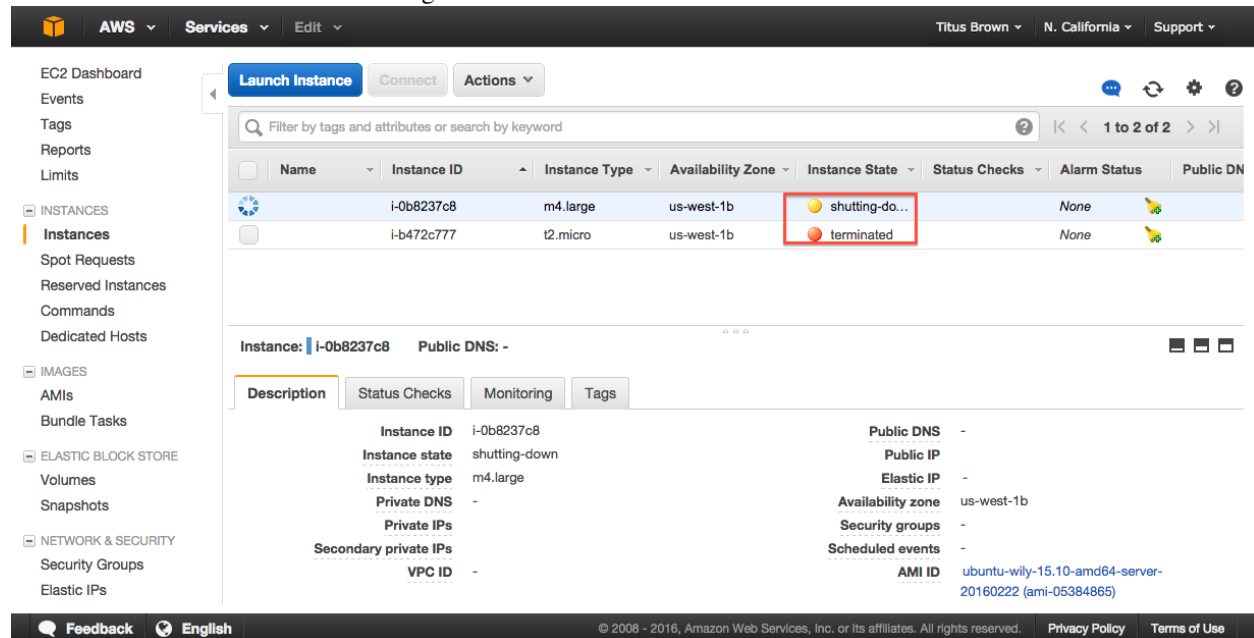

**Warning**  
 On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?  
 i-0b8237c8

Cancel
Yes, Terminate

## 3. Verify status on your instance page.

Instance state should be either “shutting down” or “terminated”.



The screenshot shows the AWS Management Console interface for EC2 instances. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area displays a table of instances. Two instances are highlighted with red boxes: i-0b8237c8 (shutting-down) and i-b472c777 (terminated). Below the table, the details for instance i-0b8237c8 are shown, including its state (shutting-down), type (m4.large), and availability zone (us-west-1b).

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DN
	i-0b8237c8	m4.large	us-west-1b	shutting-down		None	
	i-b472c777	t2.micro	us-west-1b	terminated		None	

Instance: i-0b8237c8 Public DNS: -

Description	Status Checks	Monitoring	Tags
Instance ID	i-0b8237c8		
Instance state	shutting-down		
Instance type	m4.large		
Private DNS	-		
Private IPs			
Secondary private IPs			
VPC ID	-		
Public DNS	-		
Public IP			
Elastic IP	-		
Availability zone	us-west-1b		
Security groups	-		
Scheduled events	-		
AMI ID	ubuntu-wily-15.10-amd64-server-20160222 (ami-05384865)		

Return to index: [Starting up an Amazon Web Services machine](#)

## 2.2.9 Things to mention and discuss

### When do disks go away?

- never on reboot;
- ephemeral disks go away on stop;

- AMI-attached volumes go away on terminate;
- attached volumes never go away on terminate and have to be explicitly deleted;
- snapshots only go away when you explicitly delete them.

### What are you charged for?

- you are charged for a running instance at the @@instance price rates;
- ephemeral storage/instance-specific storage is included within that.
- when you stop an instance, you are charged at disk-space rates for the stopped disk;
- when you create a volume, you are charged for that volume until you delete it;
- when you create a snapshot, you are charged for that snapshot until you delete it.

To make sure you're not getting charged, go to your Instance view and clear all search filters; anything that is "running" or "stopped" is costing you. Also check your volumes and your snapshots - they should be empty.

---

### Regions vs zones:

- AMIs and Snapshots (and keys and security groups) are per region;
- Volumes and instances are per zone;

## 2.2.10 Running RStudio Server in the cloud

In this section, we will run RStudio Server on a remote Amazon machine. This will require starting up an instance, configuring its network firewall, and installing and running some software.

Reference documentation for running RStudio Server on Ubuntu:

<https://www.rstudio.com/products/rstudio/download-server/>

---

### 1. Start up an Amazon instance

Start an ami-05384865 on an m4.xlarge machine, as per the instructions here:

[Start an Amazon Web Services computer](#).

### 2. Configure your network firewall

Normally, Amazon computers only allow shell logins via ssh. Since we want to run a Web service, we need to give the outside world access to other network locations on the computer.

Follow these instructions:

[Configure your instance firewall](#)

(You can do this while the computer is booting.)

### 3. Log in via the shell

Follow these instructions to log in via the shell:

Log into your instance with the [UNIX shell](#).

### 4. Set a password for the ‘ubuntu’ account

Amazon Web Services computers normally require a key (the .pem file) instead of a login password, but RStudio Server will need us to log in with a password. So we need to configure a password for the account we’re going to use (which is ‘ubuntu’)

Create a password like so:

```
sudo passwd ubuntu
```

and set it to something you’ll remember.

### 5. Install R and the gdebi tool

Update the software catalog and install a few things:

```
sudo apt-get update && sudo apt-get -y install gdebi-core r-base
```

This will take a few minutes.

### 6. Download & install RStudio Server

```
wget https://download2.rstudio.org/rstudio-server-0.99.891-amd64.deb
sudo gdebi -n rstudio-server-0.99.891-amd64.deb
```

Upon success, you should see:

```
Mar 07 15:20:18 ip-172-31-6-68 systemd[1]: Starting RStudio Server...
Mar 07 15:20:18 ip-172-31-6-68 systemd[1]: Started RStudio Server.
```

### 7. Open your RStudio Server instance

Finally, go to ‘[http://](#)’ + your hostname + ‘:8787’ in a browser, eg.

```
http://ec2-XX-YY-33-165.us-west-1.compute.amazonaws.com:8787/
```

and log into RStudio with username ‘ubuntu’ and the password you set it to above.

Voila!

---

You can now just go ahead and use this, or you can “stop” it, or you can freeze into an AMI for later use.

Note that on reboot, RStudio Server will start up again and all your files will be there.

Go back to the index: [Starting up an Amazon Web Services machine](#).

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



---

## Short read quality and trimming

---

Start up an instance with ami-05384865 and 500 GB of local storage ([Start an Amazon Web Services computer](#)). You should also configure your firewall ([Configure your instance firewall](#)) to pass through TCP ports 8000-8888.

Then, Log into your computer.

—  
You should now be logged into your Amazon computer! You should see something like this:

```
ubuntu@ip-172-30-1-252:~$
```

this is the command prompt.

### 4.1 Prepping the computer

Before we do anything else, we need to set up a place to work and install a few things.

First, let's set up a place to work. Here, we'll make /mnt writeable:

```
sudo chmod a+rwxt /mnt
```

---

**Note:** /mnt is the location we're going to use on Amazon computers, but if you're working on a local cluster, it will have a different location. Talk to your local sysadmin and ask them where they recommend putting lots of short-term working files, i.e. the “scratch” space.

---

### 4.2 Installing some software

Run:

```
sudo apt-get -y update && \  
sudo apt-get -y install trimmomatic fastqc python-pip \  
samtools zlib1g-dev ncurses-dev python-dev
```

Install anaconda:

```
curl -O https://repo.continuum.io/archive/Anaconda3-4.2.0-Linux-x86_64.sh  
bash Anaconda3-4.2.0-Linux-x86_64.sh
```

Then update your environment and install khmer:

```
source ~/.bashrc

cd
git clone https://github.com/dib-lab/khmer.git
cd khmer
sudo python2 setup.py install
```

## 4.3 Running Jupyter Notebook

Let's also run a Jupyter Notebook in /mnt. First, configure it a teensy bit more securely, and also have it run in the background:

```
jupyter notebook --generate-config

cat >>/home/ubuntu/.jupyter/jupyter_notebook_config.py <<EOF
c = get_config()
c.NotebookApp.ip = '*'
c.NotebookApp.open_browser = False
c.NotebookApp.password = u'sha1:5d813e5d59a7:b4e430cf6dbd1aad04838c6e9cf684f4d76e245c'
c.NotebookApp.port = 8000

EOF
```

Now, run!

```
cd /mnt
jupyter notebook &
```

You should be able to visit port 8000 on your AWS computer and see the Jupyter console. (The password is 'davis'.)

## 4.4 Data source

We're going to be using a subset of data from [Hu et al., 2016](#). This paper from the Banfield lab samples some relatively low diversity environments and finds a bunch of nearly complete genomes.

(See [DATA.md](#) for a list of the data sets we're using in this tutorial.)

## 4.5 1. Copying in some data to work with.

We've loaded subsets of the data onto an Amazon location for you, to make everything faster for today's work. We're going to put the files on your computer locally under the directory /mnt/data:

```
mkdir /mnt/data
```

Next, let's grab part of the data set:

```
cd /mnt/data
curl -O -L https://s3-us-west-1.amazonaws.com/dib-training.ucdavis.edu/metagenomics-scripps-2016-10-1
curl -O -L https://s3-us-west-1.amazonaws.com/dib-training.ucdavis.edu/metagenomics-scripps-2016-10-1
```

Now if you type:



```
ls -l
```

you should see something like:

```
total 346936
-rw-rw-r-- 1 ubuntu ubuntu 169620631 Oct 11 23:37 SRR1976948_1.fastq.gz
-rw-rw-r-- 1 ubuntu ubuntu 185636992 Oct 11 23:38 SRR1976948_2.fastq.gz
```

These are 1m read subsets of the original data, taken from the beginning of the file.

One problem with these files is that they are writeable - by default, UNIX makes things writeable by the file owner. Let's fix that before we go on any further:

```
chmod u-w *
```

We'll talk about what these files are below.

## 4.6 1. Copying data into a working location

First, make a working directory; this will be a place where you can futz around with a copy of the data without messing up your primary data:

```
mkdir /mnt/work
cd /mnt/work
```

Now, make a "virtual copy" of the data in your working directory by linking it in –

```
ln -fs /mnt/data/* .
```

These are FASTQ files – let's take a look at them:

```
less SRR1976948_1.fastq.gz
```

(use the spacebar to scroll down, and type 'q' to exit 'less')

Question:

- where does the filename come from?
- why are there 1 and 2 in the file names?

Links:

- [FASTQ Format](#)

## 4.7 2. FastQC

We're going to use [FastQC](#) to summarize the data. We already installed 'fastqc' on our computer for you.

Now, run FastQC on two files:

```
fastqc SRR1976948_1.fastq.gz
fastqc SRR1976948_2.fastq.gz
```

Now type 'ls':

```
ls -d *fastqc*
```

to list the files, and you should see:

```
SRR1976948_1_fastqc.html
SRR1976948_1_fastqc.zip
SRR1976948_2_fastqc.html
SRR1976948_2_fastqc.zip
```

You can download these files using your Jupyter Notebook console, if you like; or you can look at these copies of them:

- [SRR1976948\\_1\\_fastqc/fastqc\\_report.html](#)
- [SRR1976948\\_2\\_fastqc/fastqc\\_report.html](#)

Questions:

- What should you pay attention to in the FastQC report?
- Which is “better”, file 1 or file 2? And why?

Links:

- [FastQC](#)
- [FastQC tutorial video](#)

## 4.8 3. Trimmomatic

Now we’re going to do some trimming! We’ll be using [Trimmomatic](#), which (as with fastqc) we’ve already installed via apt-get.

The first thing we’ll need are the adapters to trim off:

```
curl -O -L http://dib-training.ucdavis.edu.s3.amazonaws.com/mRNAseq-semi-2015-03-04/TruSeq2-PE.fa
```

Now, to run Trimmomatic:

```
TrimmomaticPE SRR1976948_1.fastq.gz \
               SRR1976948_2.fastq.gz \
               SRR1976948_1.qc.fq.gz s1_se \
               SRR1976948_2.qc.fq.gz s2_se \
               ILLUMINACLIP:TruSeq2-PE.fa:2:40:15 \
               LEADING:2 TRAILING:2 \
               SLIDINGWINDOW:4:2 \
               MINLEN:25
```

You should see output that looks like this:

```
...
Input Read Pairs: 1000000 Both Surviving: 885734 (88.57%) Forward Only Surviving: 114262 (11.43%) Rev
TrimmomaticPE: Completed successfully
```

Questions:

- How do you figure out what the parameters mean?
- How do you figure out what parameters to use?
- What adapters do you use?
- What version of Trimmomatic are we using here? (And FastQC?)
- Do you think parameters are different for RNAseq and genomic data sets?

- What's with these annoyingly long and complicated filenames?
- why are we running R1 and R2 together?

For a discussion of optimal trimming strategies, see [MacManes, 2014](#) – it's about RNAseq but similar arguments should apply to metagenome assembly.

Links:

- [Trimmomatic](#)

## 4.9 4. FastQC again

Run FastQC again on the trimmed files:

```
fastqc SRR1976948_1.qc.fq.gz
fastqc SRR1976948_2.qc.fq.gz
```

And now view my copies of these files:

- [SRR1976948\\_1.qc\\_fastqc/fastqc\\_report.html](#)
- [SRR1976948\\_2.qc\\_fastqc/fastqc\\_report.html](#)

Let's take a look at the output files:

```
less SRR1976948_1.qc.fq.gz
```

(again, use spacebar to scroll, 'q' to exit less).

Questions:

- is the quality trimmed data “better” than before?
- Does it matter that you still have adapters!?

Optional: [K-mer Spectral Error Trimming](#)

Next: [Run the MEGAHIT assembler](#)



---

## K-mer Spectral Error Trimming

---

(Optional)

khmer documentation: <http://khmer.readthedocs.io/en/latest>

If you plot a k-mer abundance histogram of the samples, you'll notice something: there's an awful lot of unique (abundance=1) k-mers. These are erroneous k-mers caused by sequencing errors.

In a new Python3 Jupyter Notebook, run:

```
cd /mnt/work
```

and then

```
!abundance-dist-single.py -M 1e9 -k 21 SRR1976948_1.fastq.gz SRR1976948_1.fastq.gz.dist
```

and in another cell:

```
%matplotlib inline
import numpy
from pylab import *
dist1 = numpy.loadtxt('SRR1976948_1.fastq.gz.dist', skiprows=1, delimiter=',')
plot(dist1[:,0], dist1[:,1])
axis(xmax=50)
```

Many of these errors remain even after you do the Trimmomatic run; you can see this with:

```
!abundance-dist-single.py -M 1e9 -k 21 SRR1976948_1.qc.fq.gz SRR1976948_1.qc.fq.gz.dist
```

and then plot:

```
dist2 = numpy.loadtxt('SRR1976948_1.qc.fq.gz.dist', skiprows=1, delimiter=',')
plot(dist1[:,0], dist1[:,1], label='untrimmed')
plot(dist2[:,0], dist2[:,1], label='trimmed')
legend(loc='upper right')
axis(xmax=50)
```

This is for two reasons:

First, Trimmomatic trims based solely on the quality score, which is a statistical statement about the correctness of a base - a Q score of 30 means that, of 1000 bases with that Q score, 1 of those bases will be wrong. So, a base can have a high Q score and still be wrong! (and **many** bases will have a low Q score and still be correct)

Second, we trimmed **very** lightly - only bases that had a very low quality were removed. This was intentional because with assembly, you want to retain as much coverage as possible, and the assembler will generally figure out what the “correct” base is from the coverage.

An alternative to trimming based on the quality scores is to trim based on k-mer abundance - this is known as k-mer spectral error trimming. K-mer spectral error trimming *always* beats quality score trimming in terms of eliminating errors; e.g. look at this table from [Zhang et al., 2014](#):

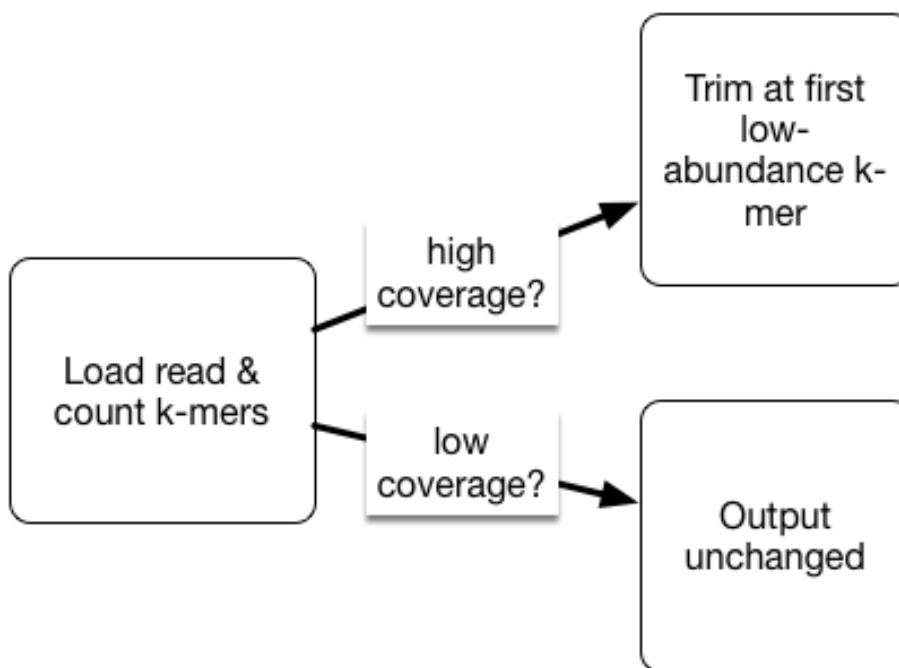
	FP rate	bases trimmed	distinct k-mers	unique k-mers
untrimmed	-	-	41.6 m	34.1 m
khmer iteration 1	80.0%	13.5%	13.3 m	6.5 m
khmer iteration 2	40.2%	1.7%	7.6 m	909.9k
khmer iteration 3	25.4%	0.3%	6.8 m	168.1k
khmer iteration 4	23.2%	0.1%	6.7 m	35.8k
khmer iteration 5	22.8%	0.0%	6.6 m	7.9k
khmer iteration 6	22.7%	0.0%	6.6 m	1.9k
filter by FASTX	-	9.1%	26.6 m	20.3 m
filter by seqtk(default)	-	8.9%	17.7 m	12.1 m
filter by seqtk(-q 0.01)	-	15.4%	9.9 m	5.1 m
filter by seqtk(-b 3 -e 5)	-	8.0%	34.5 m	27.7 m

**The results of trimming reads at unique (erroneous) k-mers from a 5 m read *E. coli* data set (1.4 GB) in under 30**

The basic logic is this: if you see low abundance k-mers in a high coverage data set, those k-mers are almost certainly the result of errors. (Caveat: strain variation could also create them.)

In metagenomic data sets we do have the problem that we may have very low and very high coverage data. So we don't necessarily want to get rid of all low-abundance k-mers, because they may represent truly low abundance (but useful) data.

As part of the khmer project in my lab, we have developed an approach that sorts reads into high abundance and low abundance reads, and only error trims the high abundance reads.



This does mean that many errors may get left in the data set, because we have no way of figuring out if they are errors or simply low coverage, but that's OK (and you can always trim them off if you really care).

To run such error trimming, use the command `trim-low-abund.py` (at the command line, or prefix with a `!` in the notebook):

```
interleave-reads.py SRR1976948_1.qc.fq.gz SRR1976948_2.qc.fq.gz |
trim-low-abund.py -V -M 8e9 -C 3 -Z 10 - -o SRR1976948.trim.fq
```

## 5.1 Why (or why not) do k-mer trimming?

If you can assemble your data set without k-mer trimming, there's no reason to do it. The reason we're error trimming here is to speed up the assembler (by removing data) and to decrease the memory requirements of the assembler (by removing a number of k-mers).

To see how many k-mers we removed, you can examine the distribution as above, or use the `unique-kmers.py` script:

```
unique-kmers.py SRR1976948_1.qc.fq.gz SRR1976948_2.qc.fq.gz
unique-kmers.py SRR1976948.trim.fq
```

---

Next: [Run the MEGAHIT assembler](#)



---

## Run the MEGAHIT assembler

---

MEGAHIT is a very fast, quite good assembler designed for metagenomes.

First, install it:

```
cd
git clone https://github.com/voutcn/megahit.git
cd megahit
make
```

Now, download some data:

```
cd /mnt/data
curl -O https://s3-us-west-1.amazonaws.com/dib-training.ucdavis.edu/metagenomics-scripps-2016-10-12/
curl -O https://s3-us-west-1.amazonaws.com/dib-training.ucdavis.edu/metagenomics-scripps-2016-10-12/
```

These are data that have been run through k-mer abundance trimming (see [K-mer Spectral Error Trimming](#)) and subsampled so that we can run an assembly in a fairly short time period :).

---

Now, finally, run the assembler!

```
mkdir /mnt/assembly
cd /mnt/assembly
ln -fs ../data/*.subset.pe.fq.gz .

~/megahit/megahit --12 SRR1976948.abundtrim.subset.pe.fq.gz,SRR1977249.abundtrim.subset.pe.fq.gz \
-o combined
```

This will take about 25 minutes; at the end you should see output like this:

```
... 12787984 bp, min 200 bp, max 61353 bp, avg 1377 bp, N50 3367 bp
... ALL DONE. Time elapsed: 1592.503825 seconds
```

The output assembly will be in `combined/final.contigs.fa`.

### 6.1 While the assembly runs...

How assembly works - whiteboarding the De Bruijn graph approach.

Interpreting the MEGAHIT working output :)

What does, and doesn't, assemble?

How good is assembly anyway?

Discussion:

Why would we assemble, vs looking at raw reads? What are the advantages and disadvantages?

What are the technology tradeoffs between Illumina HiSeq, Illumina MiSeq, and PacBio? (Also see [this paper](#).)

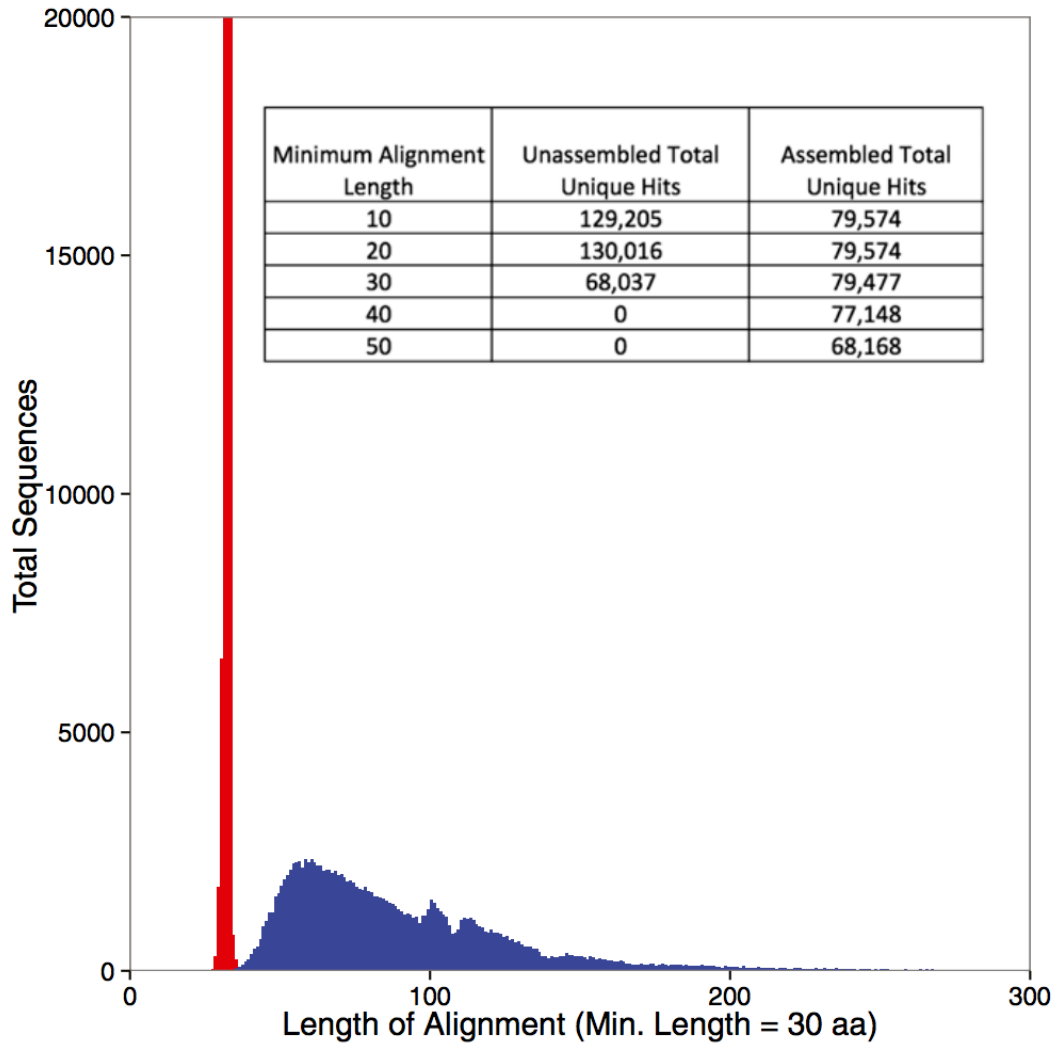
What kind of experimental design considerations should you have if you plan to assemble?

Some figures: the first two come from work by Dr. Sherine Awad on analyzing the data from Shakya et al (2014). The third comes from an analysis of read search vs contig search of a protein database.

**Table 1. Running Time and Memory Utilization**

(1) IDBA-UD	
Running Time	33h 54m
Memory Utilization (GB)	123.84
(2) SPAdes	
Running Time	67h 02m
Memory Utilization (GB)	381.79
(3) MEGAHIT	
Running Time	1h 53m
Memory Utilization (GB)	33.41

(2) Ambiguous Approach		
(1) IDBA-UD		
99.0	Genome Coverage Duplication Ratio	88.84% 0.92%
95.0	Genome Coverage Duplication Ratio	94.98% 1.84%
(2) SPAdes		
99.0	Genome Coverage Duplication Ratio	88.34% 0.83%
95.0	Genome Coverage Duplication Ratio	94.96% 1.11%
(3) MEGAHIT		
99.0	Genome Coverage Duplication Ratio	88.70% 0.10%
95.0	Genome Coverage Duplication Ratio	94.47% 1.73%



## 6.2 After the assembly is finished

At this point we can do a bunch of things:

- annotate the assembly ([Annotation with Prokka](#));
- evaluate the assembly's inclusion of k-mers and reads;
- set up a BLAST database so that we can search it for genes of interest;
- quantify the abundance of the contigs or genes in the assembly, using the original read data set ([Gene Abundance Estimation with Salmon](#));
- bin the contigs in the assembly into species bins;



---

## Annotation with Prokka

---

Prokka is a tool that facilitates the fast annotation of prokaryotic genomes.

The goals of this tutorial are to:

- Install Prokka
- Use Prokka to annotate our genomes

### 7.1 Installing Prokka

Download and extract the latest version of prokka:

```
cd ~/
wget http://www.vicbioinformatics.com/prokka-1.11.tar.gz
tar -xvzf prokka-1.11.tar.gz
```

We also will need some dependencies such as bioperl:

```
sudo apt-get install bioperl libdatetime-perl libxml-simple-perl libdigest-md5-perl
sudo perl -MCPAN -e shell
sudo perl -MCPAN -e 'install "XML::Simple"'
```

Now, you should be able to add Prokka to your \$PATH and set up the index for the sequence database:

```
export PATH=$PATH:$HOME/prokka-1.11/bin
prokka --setupdb
```

Prokka should be good to go now– you can check to make sure that all is well by typing `prokka`. This should print the help screen with all available options.

### 7.2 Running Prokka

Make a new directory for the annotation:

```
cd /mnt
mkdir annotation
cd annotation
```

Link the metagenome assembly file into this directory:

```
ln -fs /mnt/assembly/combined/final.contigs.fa
```

Now it is time to run Prokka! There are tons of different ways to specialize the running of Prokka. We are going to keep it simple for now, though. It will take a little bit to run.

```
prokka subset_assembly.fa --outdir prokka_annotation --prefix metagG
```

This will generate a new folder called `prokka_annotation` in which will be a series of files, which are detailed [here](#).

In particular, we will be using the `*.ffn` file to assess the relative read coverage within our metagenomes across the predicted genomic regions.

## 7.3 References

- <http://www.vicbioinformatics.com/software/prokka.shtml>
- <https://www.ncbi.nlm.nih.gov/pubmed/24642063>
- <https://github.com/tseemann/prokka/blob/master/README.md>

---

## Gene Abundance Estimation with Salmon

---

Salmon is one of a breed of new, very fast RNAseq counting packages. Like Kallisto and Sailfish, Salmon counts fragments without doing up-front read mapping. Salmon can be used with edgeR and others to do differential expression analysis (if you are quantifying RNAseq data).

Today we will use it to get a handle on the relative distribution of genomic reads across the predicted protein regions.

The goals of this tutorial are to:

- Install salmon
- Use salmon to estimate gene coverage in our metagenome dataset

### 8.1 Installing Salmon

Download and extract the latest version of Salmon and add it to your PATH:

```
cd
wget https://github.com/COMBINE-lab/salmon/releases/download/v0.7.2/Salmon-0.7.2_linux_x86_64.tar.gz
tar -xvzf Salmon-0.7.2_linux_x86_64.tar.gz
cd Salmon-0.7.2_linux_x86_64
export PATH=$PATH:$HOME/Salmon-0.7.2_linux_x86_64/bin
```

### 8.2 Running Salmon

Go to the data directory and download the prokka annotated sequences, assembled metagenome, and fastq files:

```
mkdir -p /mnt/data
cd /mnt/data
curl -L -O https://s3-us-west-1.amazonaws.com/dib-training.ucdavis.edu/metagenomics-scripps-2016-10-1
curl -L -O https://s3-us-west-1.amazonaws.com/dib-training.ucdavis.edu/metagenomics-scripps-2016-10-1
curl -L -O https://s3-us-west-1.amazonaws.com/dib-training.ucdavis.edu/metagenomics-scripps-2016-10-1
tar -xvzf prokka_annotation_assembly.tar.gz
```

Make a new directory for the quantification of data with Salmon:

```
mkdir /mnt/quant
cd /mnt/quant
```

Grab the nucleotide (\*ffn) predicted protein regions from Prokka and link them here. Also grab the trimmed sequence data (\*fq)

```
ln -fs /mnt/data/prokka_annotation/*.ffn .
ln -fs /mnt/data/*.abundtrim.subset.pe.fq.gz .
```

Create the salmon index:

```
salmon index -t metagG.ffn -i transcript_index --type quasi -k 31
```

Salmon requires that paired reads be separated into two files. We can split the reads using the `split-paired-reads.py` from the khmer package:

```
for file in *.abundtrim.subset.pe.fq.gz
do
    tail=.fq.gz
    BASE=${file/$tail/}
    split-paired-reads.py $BASE$tail -1 ${file/$tail/}.1.fq -2 ${file/$tail/}.2.fq
done
```

Now, we can quantify our reads against this reference:

```
for file in *.pe.1.fq
do
    tail1=.abundtrim.subset.pe.1.fq
    tail2=.abundtrim.subset.pe.2.fq
    BASE=${file/$tail1/}
    salmon quant -i transcript_index --libType IU \
        -1 $BASE$tail1 -2 $BASE$tail2 -o $BASE.quant;
done
```

(Note that `--libType` must come before the read files!)

This will create a bunch of directories named after the fastq files that we just pushed through. Take a look at what files there are within one of these directories:

```
find SRR1976948.quant -type f
```

## 8.3 Working with count data

Now, the `quant.sf` files actually contain the relevant information about expression – take a look:

```
head -10 SRR1976948.quant/quant.sf
```

The first column contains the transcript names, and the fourth column is what we will want down the road - the normalized counts (TPM). However, they're not in a convenient location / format for use; let's fix that.

Download the `gather-counts.py` script:

```
curl -L -O https://raw.githubusercontent.com/ngs-docs/2016-metagenomics-sio/master/gather-counts.py
```

and run it:

```
python2 ./gather-counts.py
```

This will give you a bunch of `.counts` files, which are processed from the `quant.sf` files and named for the directory from which they emanate.



## 8.4 Plotting the results

In Jupyter Notebook, open a new Python3 notebook and enter:

```
%matplotlib inline
import numpy
from pylab import *
```

In another cell:

```
cd /mnt/quant
```

In another cell:

```
counts1 = [ x.split()[1] for x in open('SRR1976948.quant.counts')]
counts1 = [ float(x) for x in counts1[1:] ]
counts1 = numpy.array(counts1)

counts2 = [ x.split()[1] for x in open('SRR1977249.quant.counts')]
counts2 = [ float(x) for x in counts2[1:] ]
counts2 = numpy.array(counts2)

plot(counts1, counts2, '*')
```

## 8.5 References

- <http://salmon.readthedocs.io/en/latest/salmon.html>
- <http://biorxiv.org/content/early/2016/08/30/021592>



---

## Day 2 - installation instructions

---

(Instructions mostly copied from [Short read quality and trimming!](#))

Use ami-05384865, with a 500 GB local disk (see: [Start an Amazon Web Services computer:](#))

Make /mnt/ read/write:

```
sudo chmod a+rwxt /mnt
```

Run:

```
sudo apt-get -y update && \  
sudo apt-get -y install trimmomatic fastqc python-pip \  
    samtools zlib1g-dev ncurses-dev python-dev
```

Install anaconda:

```
curl -O https://repo.continuum.io/archive/Anaconda3-4.2.0-Linux-x86_64.sh  
bash Anaconda3-4.2.0-Linux-x86_64.sh
```

Then update your environment and install khmer:

```
source ~/.bashrc  
  
cd  
git clone https://github.com/dib-lab/khmer.git  
cd khmer  
sudo python2 setup.py install
```

### 9.1 Running Jupyter Notebook

Let's also run a Jupyter Notebook in /mnt. First, configure it a teensy bit more securely, and also have it run in the background.

Generate a config:

```
jupyter notebook --generate-config
```

Add a password, have it not run a browser, and put it on port 8000 by default:

```
cat >>/home/ubuntu/.jupyter/jupyter_notebook_config.py <<EOF  
c = get_config()  
c.NotebookApp.ip = '*'  
c.NotebookApp.open_browser = False
```

```
c.NotebookApp.password = u'sha1:5d813e5d59a7:b4e430cf6dbd1aad04838c6e9cf684f4d76e245c'
c.NotebookApp.port = 8000

EOF
```

Now, run!

```
cd /mnt
jupyter notebook &
```

You should be able to visit port 8000 on your AWS computer and see the Jupyter console.

---

## Technical information

---

The github repository for this workshop is public at <https://github.com/ngs-docs/2016-metagenomics-sio>