L01: Massive Data and AWS

ANLY 502: Massive Data Fundamentals

Simson Garfinkel & Marck Vaisman

January 11, 2017

Please fill out survey at http://bit.ly/anly502-2017

17 89

GEORGETOWN UNIVERSITY

Simson Garfinkel

Marck Vaisman

Welcome!

Q: What is “massive data?”

(think about it!)

(Please fill out survey at http://bit.ly/ANLY502-2016)

ANLY 502 — Massive Data Fundamentals (Also COSC-588)

Overview

- ““Today's data scientists are commonly faced with huge data sets (Big Data) that may arrive at fantastic rates and in a broad variety of formats. This core course addresses the resulting challenges to data professionals. The course will introduce students to the advantages and limitations of distributed computing and to methods of assessing its impact. Techniques for parallel processing (MapReduce) and their implementation (Hadoop) will be covered, as well as techniques for accessing unstructured data and for handling streaming data. These techniques will be applied to real world examples, using clusters of computational cores and cloud computing. Prerequisite: Good command of R or Python, some knowledge of data structures. Three credits”

Spring 2017 • Mon. 6:30 — 9:00 (except tonight, which is Wed.)

This is class is 1 year old!

- This is our first time teaching together!
- This class is designed to be forward-looking and research-focused

Before we get started, please fill out the class survey:

- bit.ly/ANLY502-2017

Introducing your teachers.

Simson L. Garfinkel, Ph.D.
US Census Bureau*
https://simson.net/
sg1224@georgetown.edu

Interests: Security, Privacy, Digital Forensics

Marck Vaisman
Booz Allen Hamilton, Data Community DC*
mv559@georgetown.edu

Interests: Data Science in Business, High Performance Computing, R development

*Institutional affiliation is provided for identification purposes only.

Outline for today's class

Introduction to ANLY 502

- Course introduction, policies and outline
- What you need to succeed in ANLY 502
- Information about labs and Amazon

Massive Data and the end of "Moore's law"

- Where will tomorrow's computing speed increases come from?

The Datacenter is the Computer — Introducing Amazon Cloud Services

Lab:

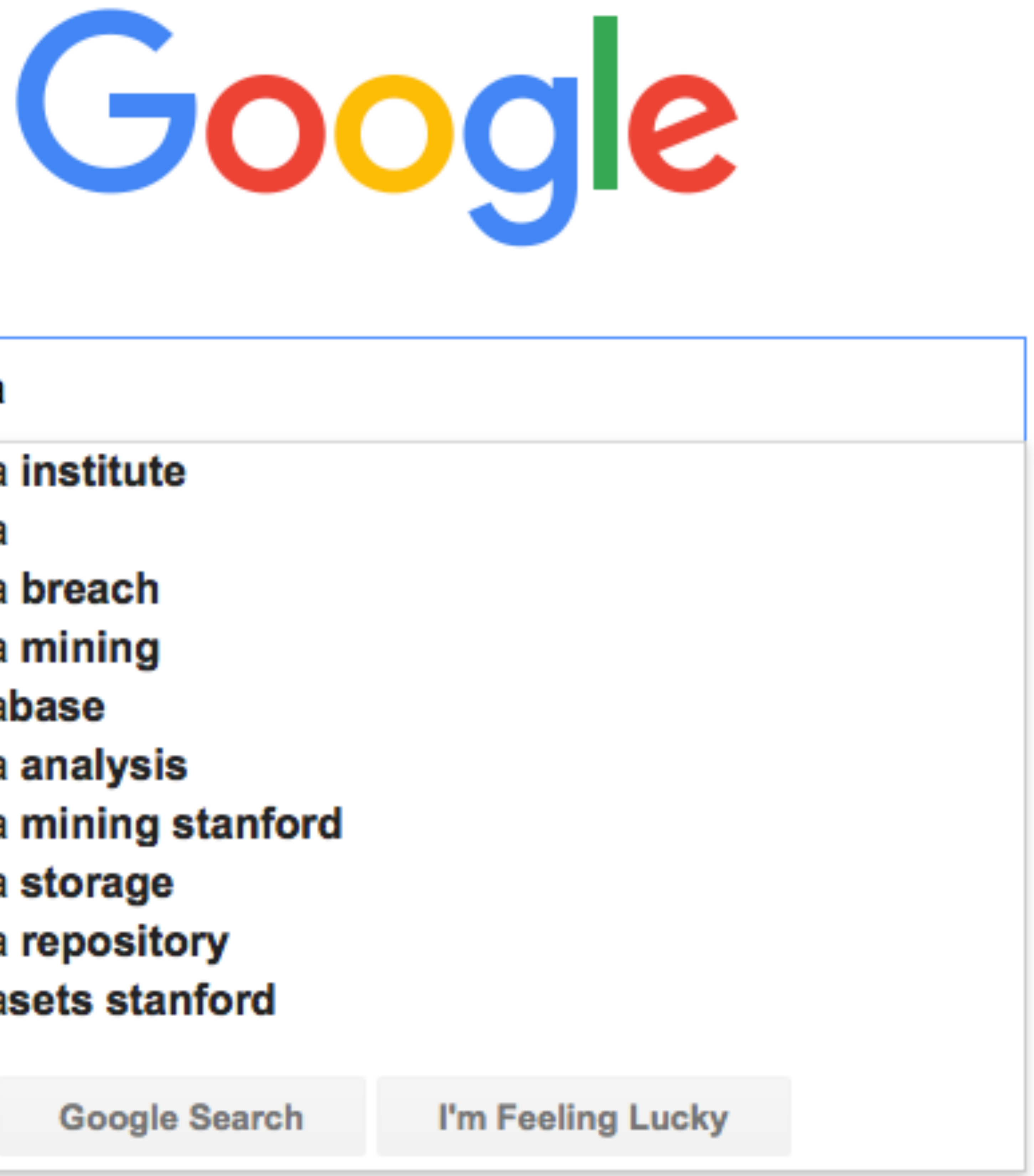
- Setting up your laptop to access Amazon

Welcome to ANLY 502: Massive Data Fundamentals

So what is “massive data?”

So what is massive data?

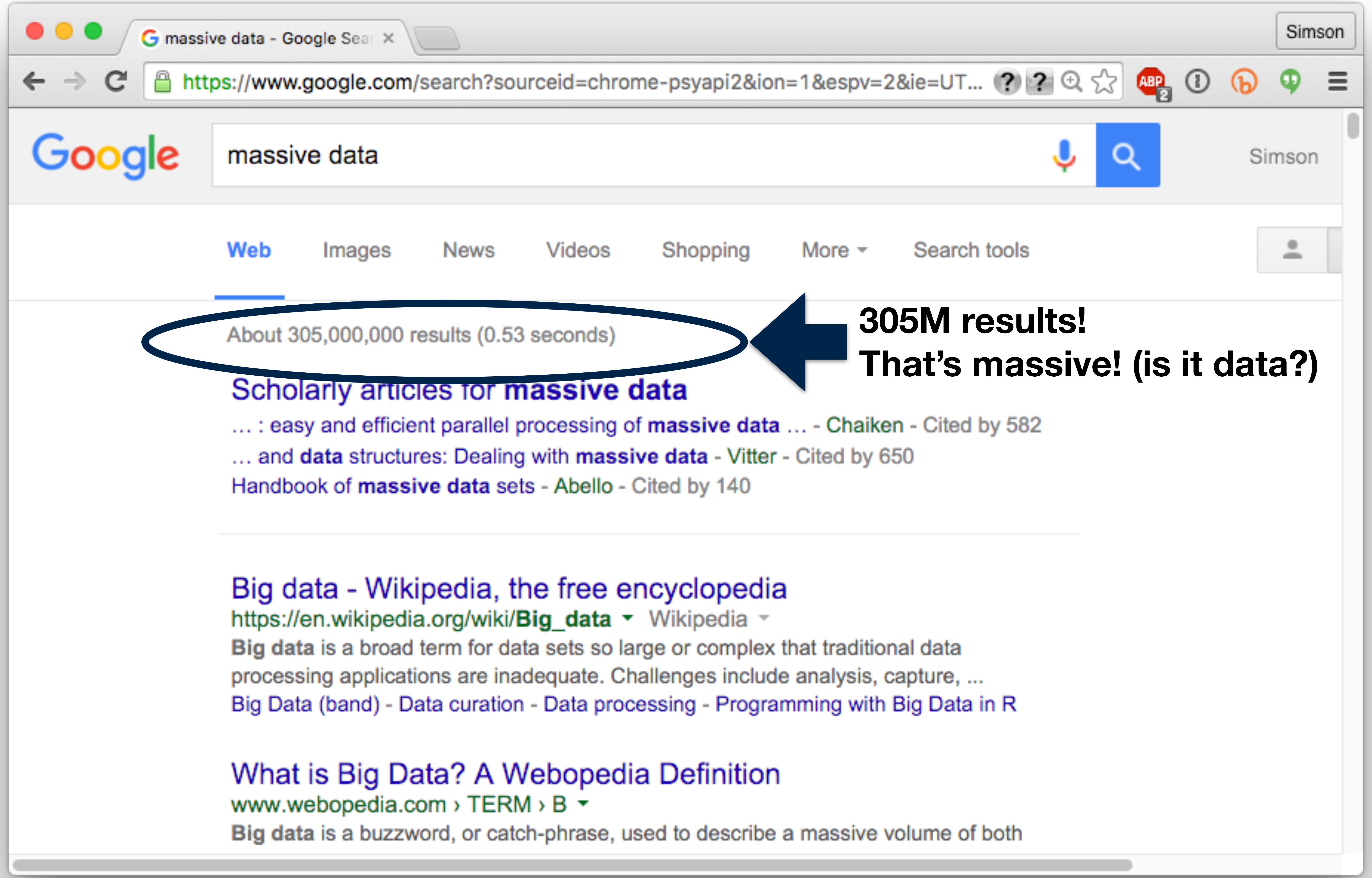
Let's ask Google:



massive data

massive data **institute**
massive data
massive data **breach**
massive data **mining**
massive data**base**
massive data **analysis**
massive data **mining stanford**
massive data **storage**
massive data **repository**
massive data**sets stanford**

Google's view of massive data:



massive data - Google Sea ×
Simson
https://www.google.com/search?sourceid=chrome-psyapi2&ion=1&espv=2&ie=UT...
Google
massive data
Simson
Web
Images
News
Videos
Shopping
More
Search tools

About 305,000,000 results (0.53 seconds)

Scholarly articles for **massive data**

… : easy and efficient parallel processing of **massive data** … - Chaiken - Cited by 582

… and **data** structures: Dealing with **massive data** - Vitter - Cited by 650

Handbook of **massive data** sets - Abello - Cited by 140

Big data - Wikipedia, the free encyclopedia

https://en.wikipedia.org/wiki/**Big_data** ▾ Wikipedia ▾

Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, ...

Big Data (band) - Data curation - Data processing - Programming with Big Data in R

What is Big Data? A Webopedia Definition

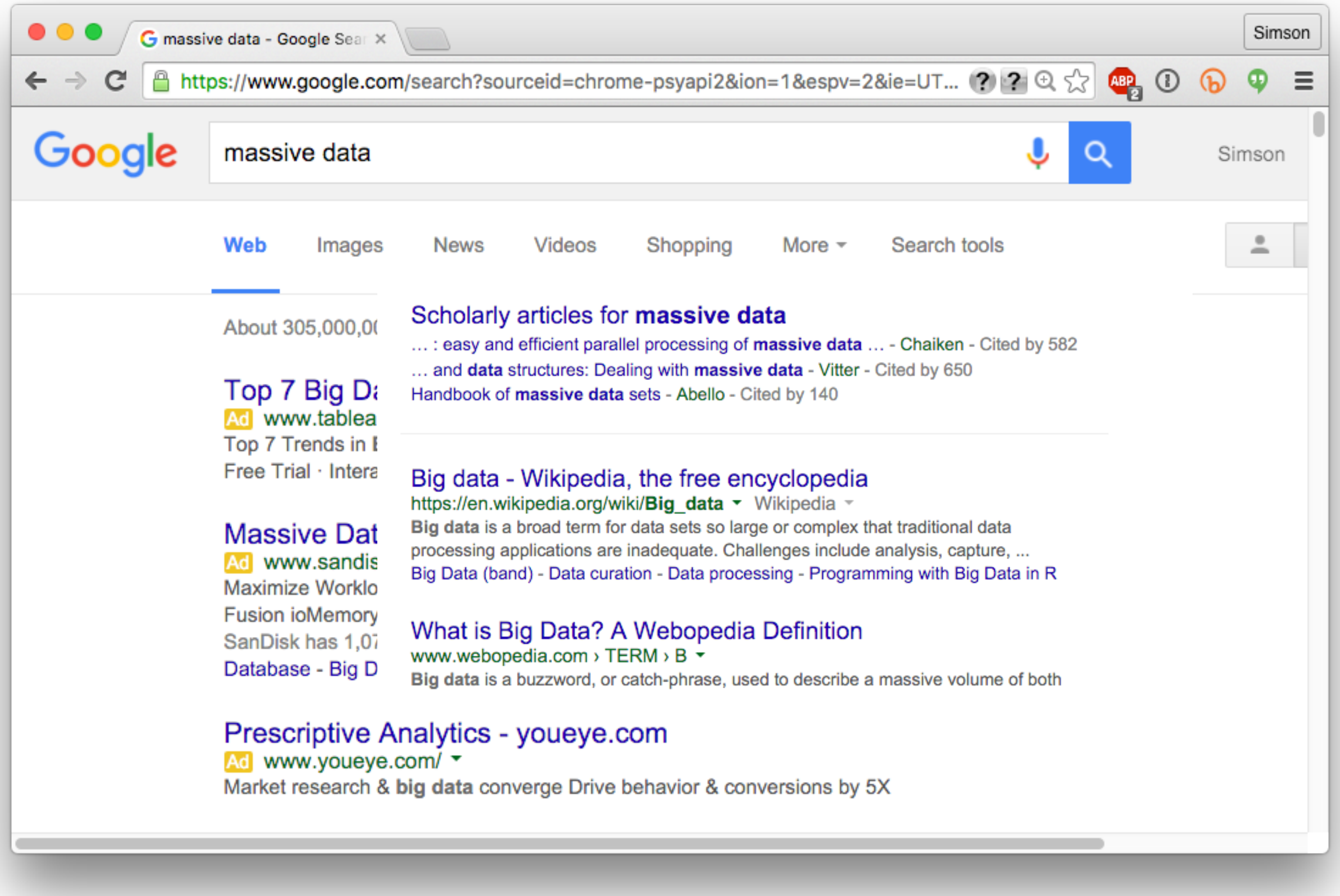
www.webopedia.com › TERM › B ▾

Big data is a buzzword, or catch-phrase, used to describe a massive volume of both

305M results!
That's massive! (is it data?)

Search results depend upon massive data.

305M results



https://www.google.com/search?sourceid=chrome-psyapi2&ion=1&espv=2&ie=UT...

massive data

About 305,000,0

Scholarly articles for **massive data**
… : easy and efficient parallel processing of **massive data** … - Chaiken - Cited by 582
… and **data** structures: Dealing with **massive data** - Vitter - Cited by 650
Handbook of **massive data** sets - Abello - Cited by 140

Top 7 Big D
Ad www.tablea
Top 7 Trends in
Free Trial · Intera

Big data - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Big_data** · Wikipedia
Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, ...
Big Data (band) - Data curation - Data processing - Programming with Big Data in R

Massive Dat
Ad www.sandis
Maximize Worklo
Fusion ioMemory
SanDisk has 1,07
Database - Big D

What is Big Data? A Webopedia Definition
www.webopedia.com › TERM › B
Big data is a buzzword, or catch-phrase, used to describe a massive volume of both

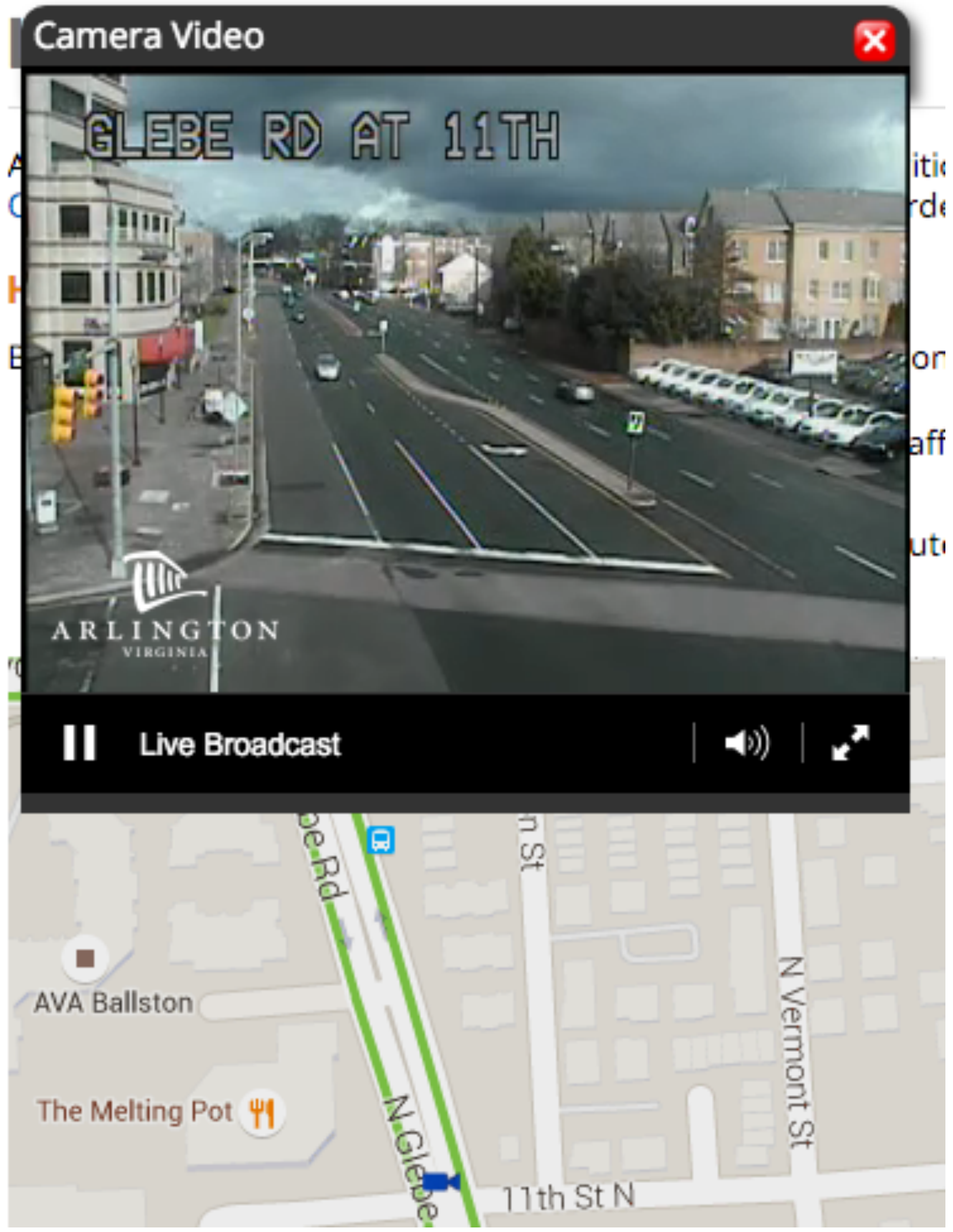
Prescriptive Analytics - youeye.com
Ad www.youeye.com/
Market research & **big data** converge Drive behavior & conversions by 5X

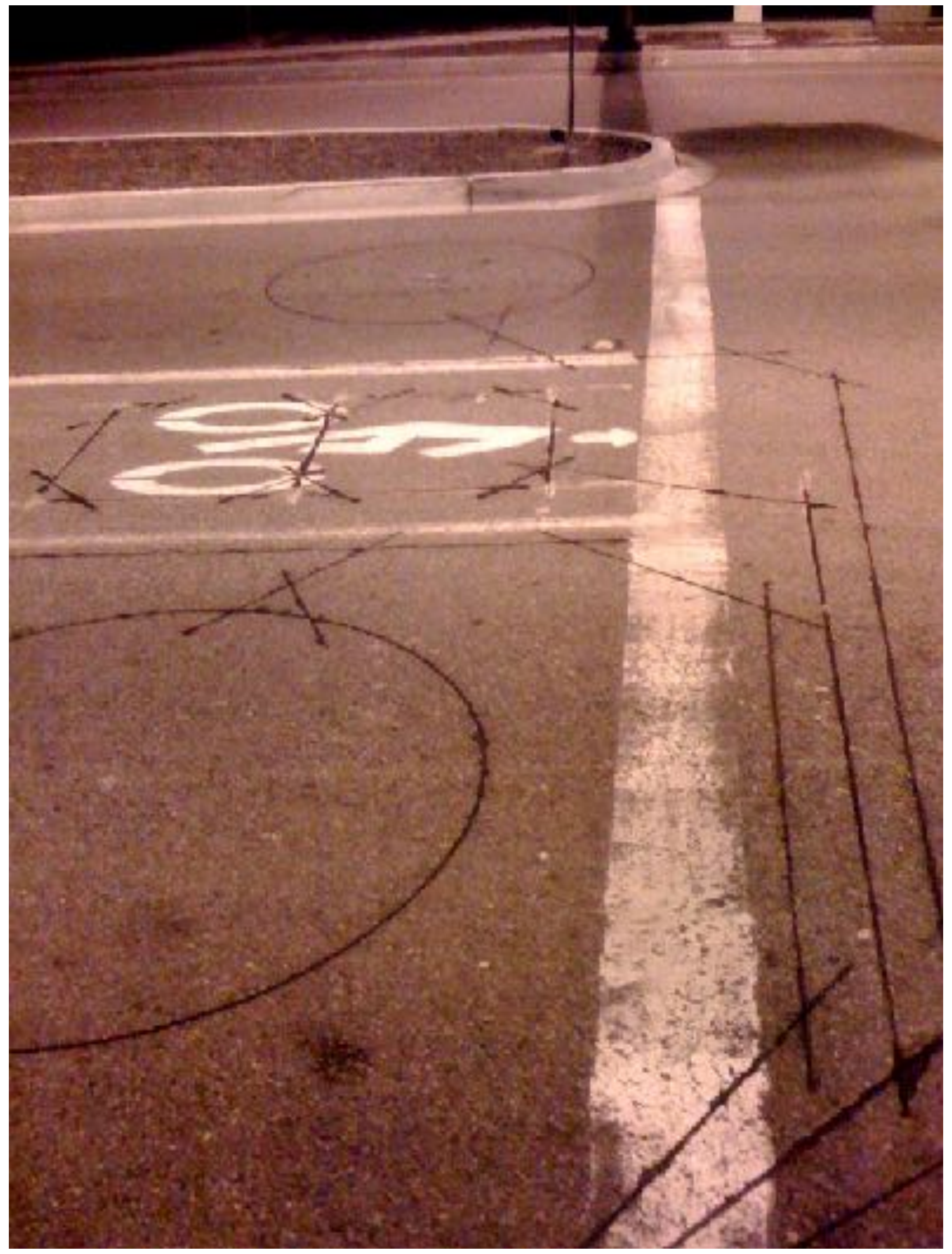
To make these results:

- Scan and index billions of web pages.
- Find all of the pages about “massive data”. (What does the word “about” mean?)
- Eliminate “spam” pages.
- Group similar pages.
- Perform search of index with billions of entries in less than a second.

Other examples of "massive data" — Real Time Traffic

Back in the 1990s, we thought real-time video over the web was "massive data."





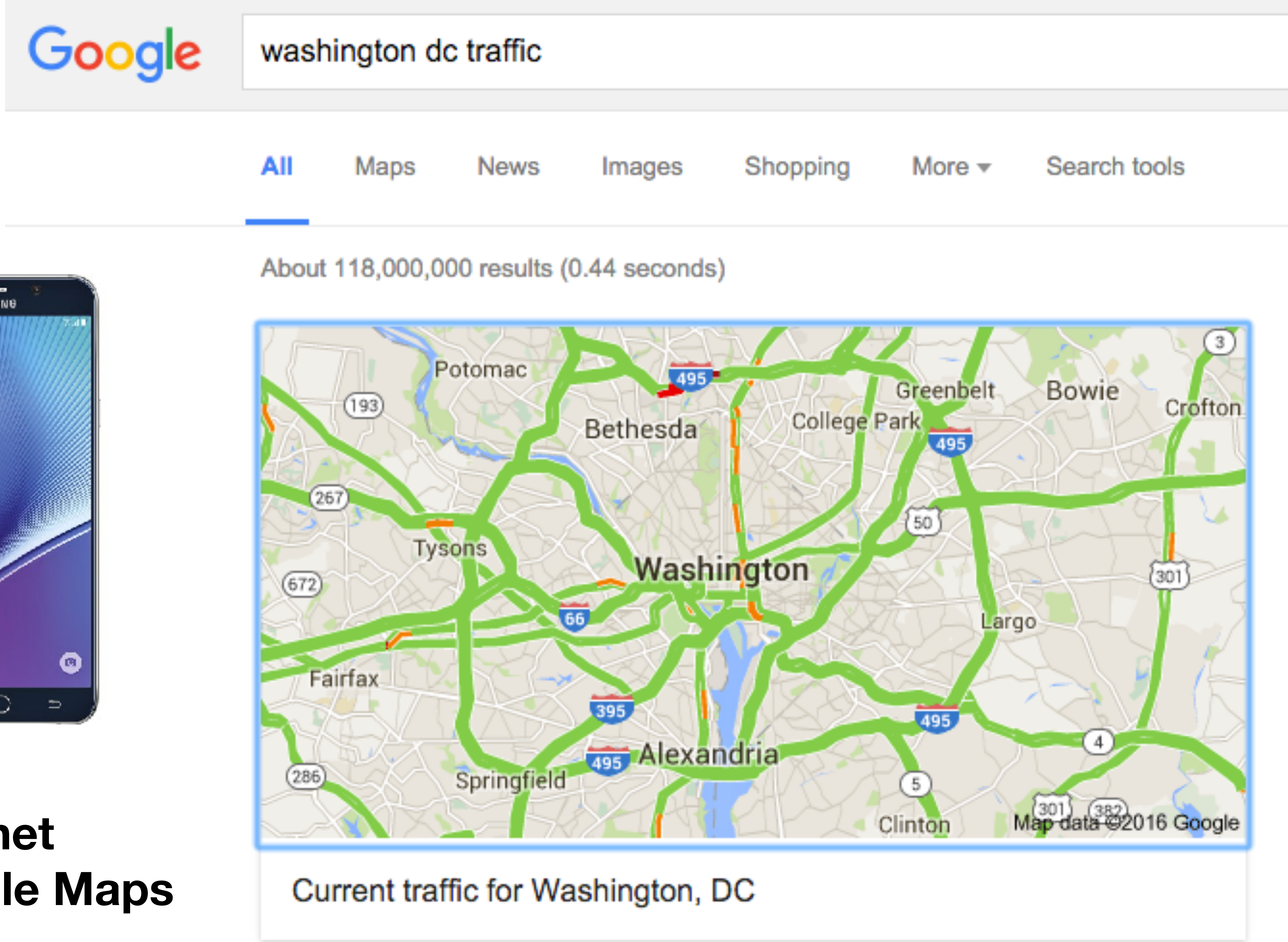
https://en.wikipedia.org/wiki/Induction_loop

Old approach to traffic: Traffic cameras and induction loops

Other examples of "massive data" — Real Time Traffic

Today's "massive data" — Using millions of cell phones as mobile traffic sensors

GPS
Internet
Google Maps

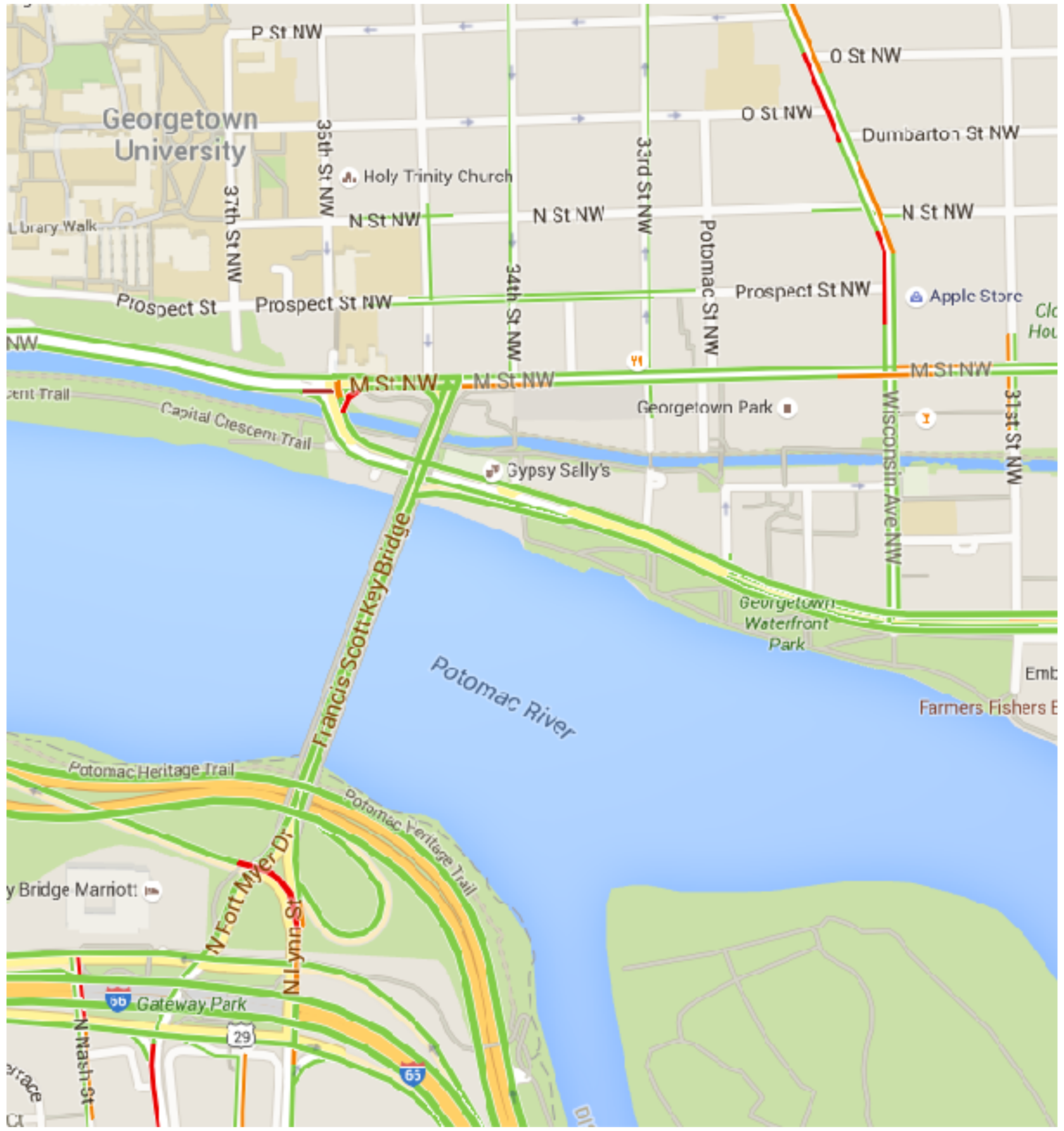


washington dc traffic

About 118,000,000 results (0.44 seconds)

Map data ©2016 Google

Current traffic for Washington, DC



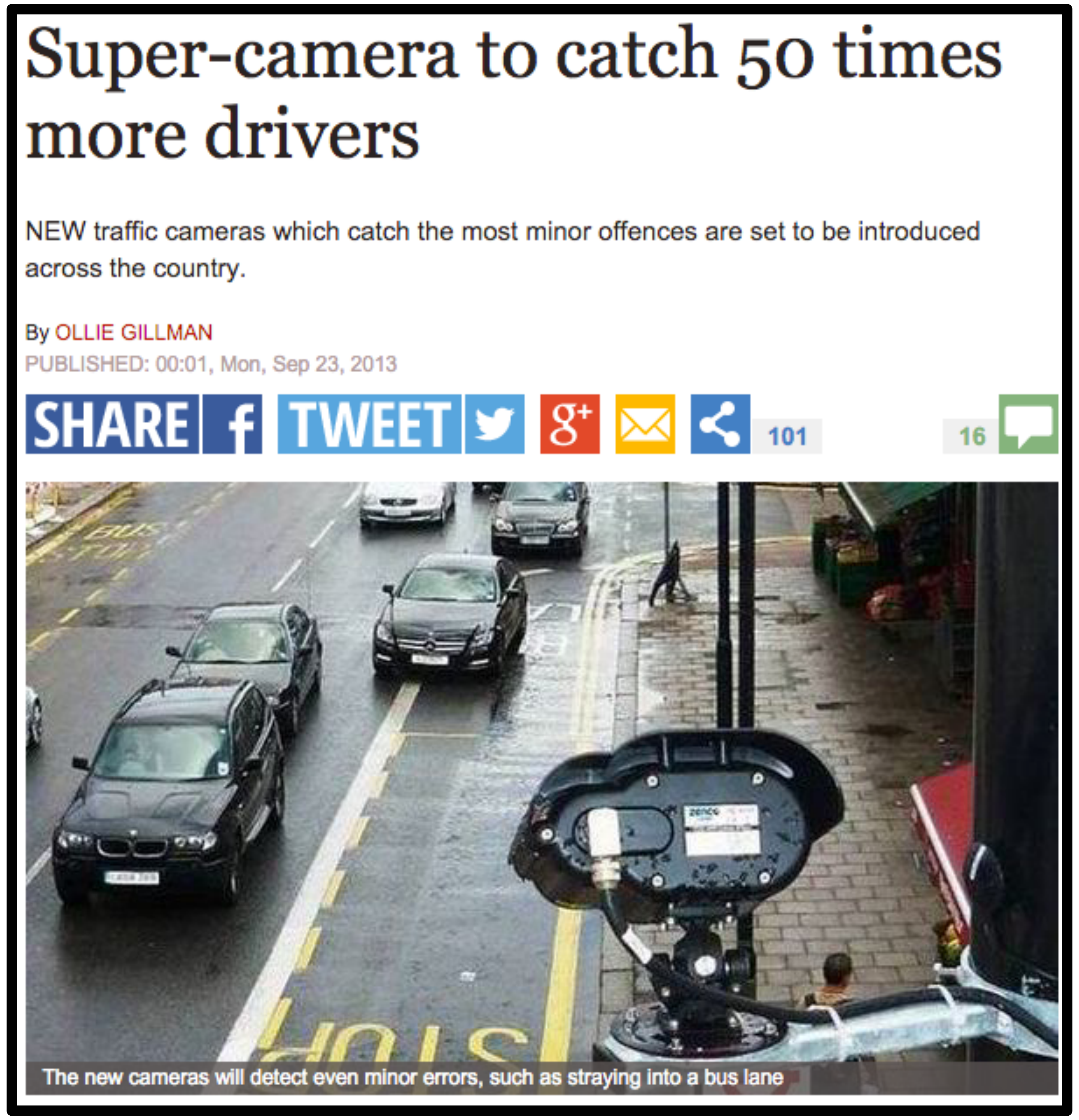
Street Level Detail

Massive data creates the potential for massive privacy problems

https://en.wikipedia.org/wiki/Traffic_enforcement_camera

http://www.cleveland.com/roadrant/index.ssf/2010/11/voters_oust_traffic_cameras_in.html

"Voters oust traffic cameras..."
Cleveland Plain Dealer
Sept. 7, 2010



Super-camera to catch 50 times more drivers

NEW traffic cameras which catch the most minor offences are set to be introduced across the country.

By OLLIE GILLMAN
PUBLISHED: 00:01, Mon, Sep 23, 2013

The new cameras will detect even minor errors, such as straying into a bus lane

http://www.express.co.uk/news/uk/431426/Super-camera-to-catch-50-times-more-drivers

Why study massive data?

Better understanding:

- Unlock truths of the past and present
- Predict the future.

Improve society and the planet:

- Public health
- Environmental monitoring & mitigation
- “Data for good” — e.g. Facebook demographics
- Cybersecurity

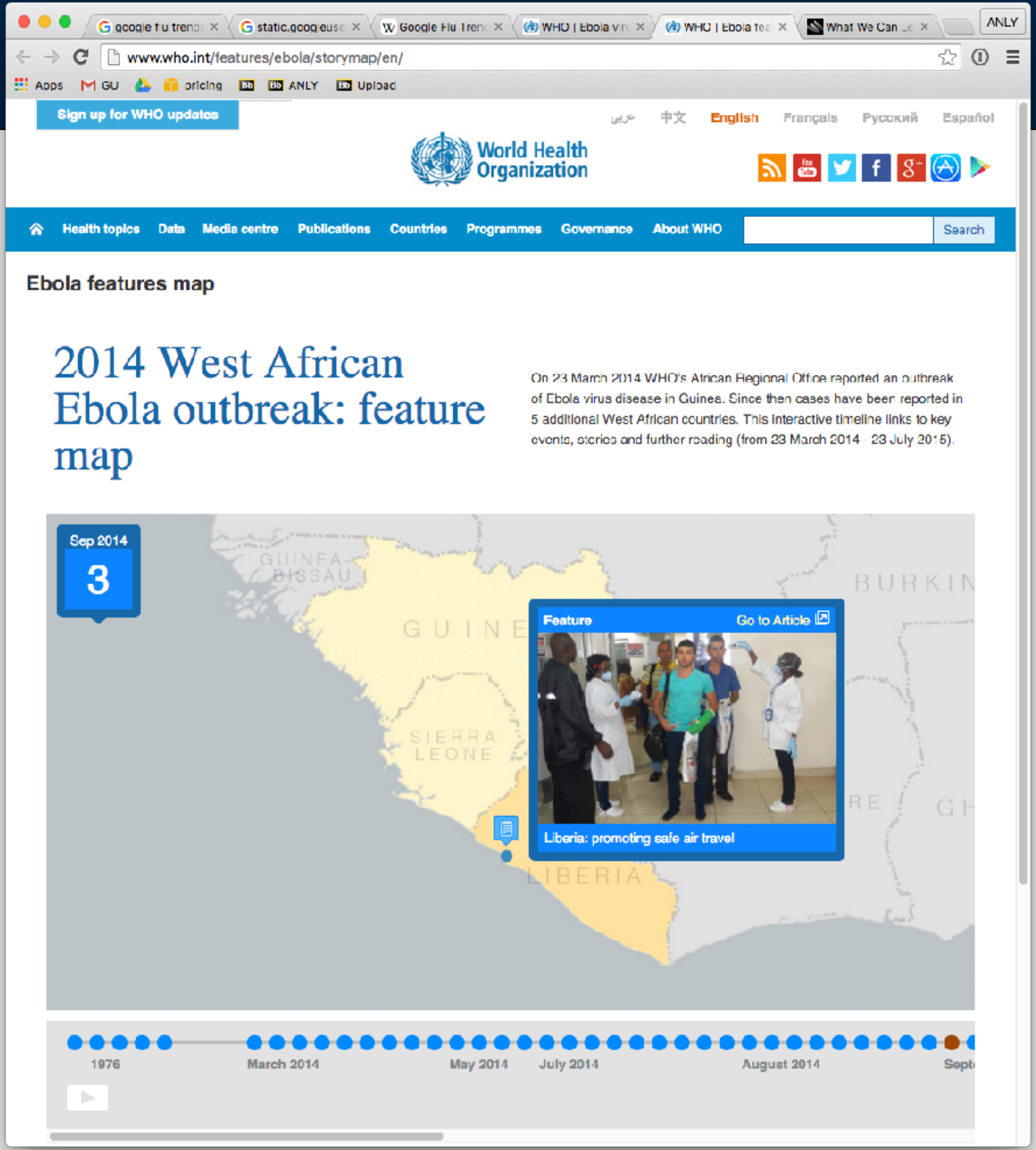
We have a data-oriented economy

- We are surrounded by data collectors.
- It’s much easier to collect data than to analyze it.
- We should be able to do *something* with all this data.

fitbit

nest

starbucks



www.who.int/features/ebola/storymap/en/

Ebola features map

2014 West African Ebola outbreak: feature map

On 23 March 2014 WHO's African Regional Office reported an outbreak of Ebola virus disease in Guinea. Since then cases have been reported in 5 additional West African countries. This interactive timeline links to key events, stories and further reading (from 23 March 2014 - 23 July 2015).

Liberia: promoting safe air travel

“Learning Outcomes”
At the end of this course, you will be able to:

Identify technical and social trends in the creation, collection, analysis and storage of massive data.

Design, cost, and assemble cloud-based computational infrastructure required to perform massive data analysis.

Perform large-scale data analysis with Python on high-performance workstations using multithreading/multiprocessing and clusters using Hadoop, Map Reduce, Apache Spark, and other advanced technologies.

Locate, download, “wrangle,” and query structured and unstructured data from Internet sources.

Research and present information about a new “Big Data” tool on the Internet.

Understand and discuss academic papers about big data technology and related social issues.

This course also introduces us to teaching about "massive data!"

Both of us have been working with Big Data for years.

- This is the first time that we've taught this course.
- This is the fist time we've worked together!

Simson L. Garfinkel, Ph.D.

Started working with "Big Data" in 1985
(Made the second CDROM in the US: 600MB of Data. Massive!

Created digital forensics data sets 500GB — 200TB in size

Developed software for forensics processing on 64-core workstations and 2000-core clusters.

Marck Vaisman

Started working with "Big Data" in 2010

Founder of DC Data Community
Specialist in big data applications and customer solutions.

ANLY 502, by the numbers

Statistics:	
First year ANLY 502 taught:	2016
Class sessions:	13
Class length:	2 hours, 40 minutes
Weeks missed because of holidays:	3
Enrolled students:	38 (as of Jan 11, 2017)
Deliverables:	
Homework assignments:	6
Final Projects:	1
Students Per project group:	3-4
Online participation:	Weekly

What you need to take this course

From the catalog:

- Prerequisites: “Good command of R or Python, some knowledge of data structures.”

Additional:

- Ability to read and write Python* code.
- Familiarity with the Unix command line and a text editor (e.g. EMACS, vi, nano, etc.)
- Commitment to homework and *working beyond the assignments*
- Access to massive data infrastructure (we’ve got this covered!)

Hardware:

- Your laptop (Mac, Linux, or Windows w/ Cygwin)
- Amazon Web Services (AWS) for hands-on “big data” work.

*Most big-data work is done with Python 2.7 due to legacy issues

Class Deliverables — What you need to do!

5 Assignments (45%)

A1 — Getting Started with AWS and Virtualization	released L01 (Jan 11); due Jan 20
A2 — Introducing MapReduce with S3 and HDFS	released L02 (Jan 23); due Feb 3
A3 — Data Wrangling and MapReduce Design Patterns	released L04 (Feb 6); due Feb 17
A4 — Distributed Analysis Patterns with Hadoop and Spark	released L06 (Feb 27); due Mar 17
A5 — SparkSQL and Pig and Possibly Hive	released L08 (Mar 20); due Mar 31

11 Quizzes — Every Friday (22%)

- Typically 5-10 multiple-choice questions

1 Final Project — Group Project involving massive data analysis (24%)

- Proposal (2%) & “Clinic” (2%)
- A presentation about your project (7%)
- A paper describing what you did (15%)

Class participation (5%) — both in class and online

Final project sequence and timeline

Tue. Mar 22 — Final Project Individual Proposals Due

- Each student must write two proposals (1 paragraph each)
- Proposals must be posted in the form on Canvas.

Tue. March 28 — Final project group proposals due

- Each group must submit a 1-2 page proposal clearly documenting what will be done, by whom, with a timeline.

Mon. April 3rd — Proposal response: “accepted” or “revise”

Mon. April 10–April 14 — Final Project Online “Clinic”

- April 12 — April 18 — Easter Break

Mon., May 1 — Final projects presented in class

Wed, May 10 — Final projects paper due

Make the most of your final project — It's a quarter of your grade!

Your final project must include:

- Literature review
- Clear contribution — data analysis, tool development, etc.
- Validation — how do you prove that you did what you said you did?
- Conclusion
  - *Start thinking about your final projects now!*

Your final project deliverables include:

- Proposals 1 & 2
- A paper with an abstract, background, literature search, main body, and conclusion
- A slide presentation
- Optional video demo:
  - *Demos should be 60-120 seconds of video*
  - *Demos should be uploaded to Black Board or YouTube. Fri., March 25 — Final project individual proposals due*

Required Readings & Optional Readings

Readings are associated with every class

- So we can discuss them in class.

Readings should be completed before class starts!

You are responsible for the content of the required readings.

- You will not be tested on readings that are not discussed in class.
- You may be tested on important aspects of the readings that are not explicitly discussed.

Each lesson may have one or more “optional” readings.

—*These readings are for your personal edification.*

—*Please let the class know if you find them interesting.*

—*They are pre-approved for presentations!*

Class Style

Class meets 6:30 — 9:00

- Typically class will involve:
  - *Introduction to the day*
  - *Discussion of reading & technology*
  - *Break*
  - *Lab work / problem sets / projects*

Preferred contacts:

- Email doesn't scale—Post your questions online.
  - *We will see them and answer them within 36 hours*
  - *Other students can answer as well! (please!)*
- Use email for administrative issues
  - Grades, late assignments, etc.

Please bring your laptops to class!

Class Materials

Class materials on Canvas:

- Calendar
- Discussion boards
- Assignment submission
- You can sync to Google Calendar / Phone / etc.

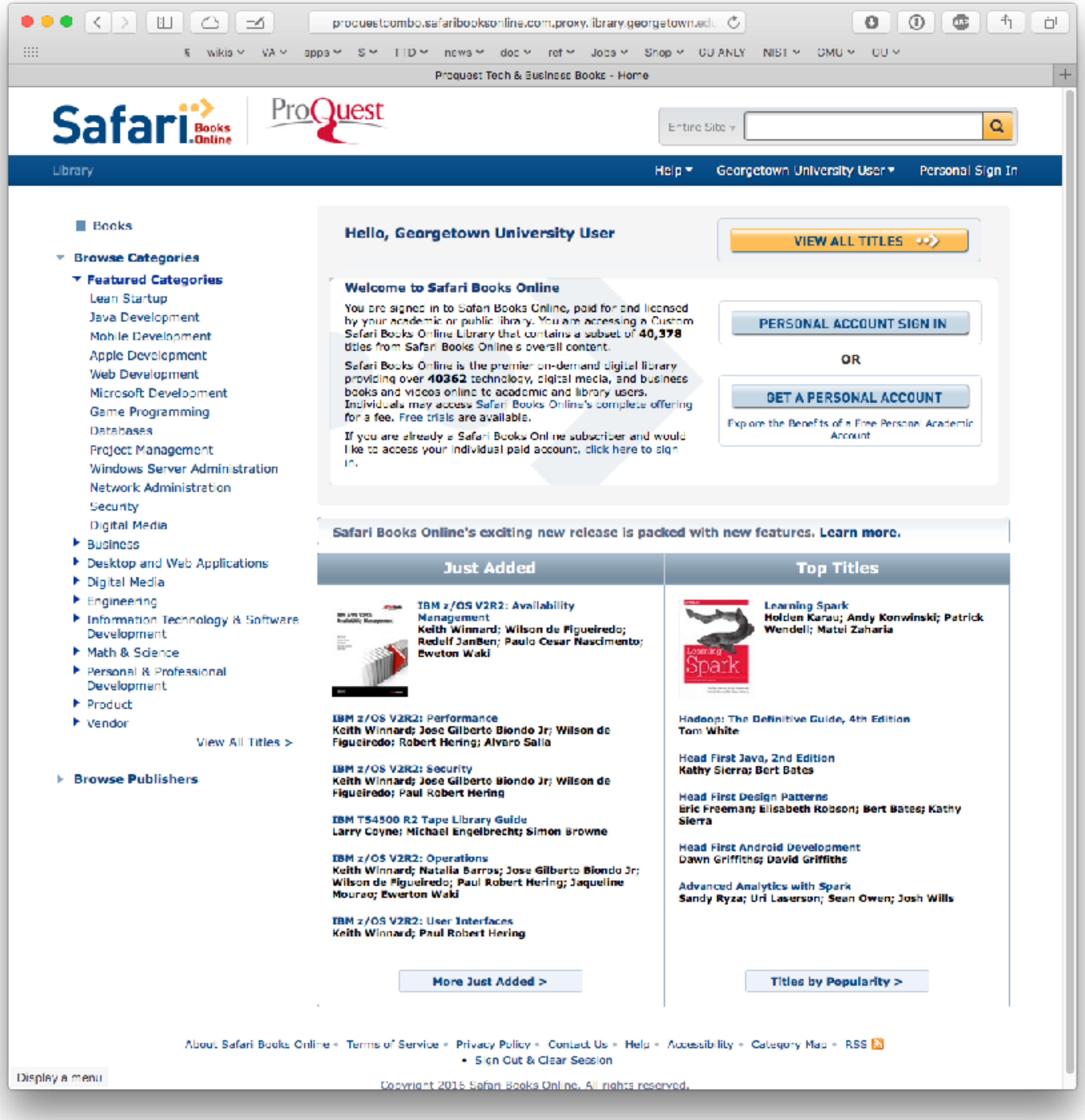
Class materials on Google Drive:

- Slides (Apple Keynote & PDF)
- Papers

Class materials on Git (BitBucket)

- Problem sets
- Starter code for problem sets
- Submission system (creates a ZIP file)
- **Fork** into a private repository

O'Reilly books are available on Safari


proquestcombo.safaribooksonline.com.proxy.library.georgetown.edu
wikis VA apps S TTD news doc ref Jobs Shop GU ANLY NIST CMU GU
Proquest Tech & Business Books - Home
Safari Books Online
ProQuest
Entire Site
Library
Help
Georgetown University User
Personal Sign In
VIEW ALL TITLES
PERSONAL ACCOUNT SIGN IN
OR
GET A PERSONAL ACCOUNT
Display a menu

Books

Browse Categories

Featured Categories

- Lean Startup
- Java Development
- Mobile Development
- Apple Development
- Web Development
- Microsoft Development
- Game Programming
- Databases
- Project Management
- Windows Server Administration
- Network Administration
- Security
- Digital Media

- Business
- Desktop and Web Applications
- Digital Media
- Engineering
- Information Technology & Software Development
- Math & Science
- Personal & Professional Development
- Product
- Vendor

View All Titles >

Browse Publishers

Hello, Georgetown University User

Welcome to Safari Books Online

You are signed in to Safari Books Online, paid for and licensed by your academic or public library. You are accessing a Custom Safari Books Online Library that contains a subset of **40,378** titles from Safari Books Online's overall content.

Safari Books Online is the premier on-demand digital library providing over **40362** technology, digital media, and business books and videos online to academic and library users. Individuals may access Safari Books Online's complete offering for a fee. Free trials are available.

If you are already a Safari Books Online subscriber and would like to access your individual paid account, click here to sign in.

Explore the Benefits of a Free Personal Academic Account

Safari Books Online's exciting new release is packed with new features. **Learn more.**

Just Added

IBM z/OS V2R2: Availability Management
Keith Winnard; Wilson de Figueiredo; Redelf JanBen; Paulo Cesar Nascimento; Eweton Waki

IBM z/OS V2R2: Performance
Keith Winnard; Jose Gilberto Biondo Jr; Wilson de Figueiredo: Robert Hering; Alvaro Salla

IBM z/OS V2R2: Security
Keith Winnard; Jose Gilberto Biondo Jr; Wilson de Figueiredo; Paul Robert Hering

IBM TS4500 R2 Tape Library Guide
Larry Coyne; Michael Engelbrecht; Simon Browne

IBM z/OS V2R2: Operations
Keith Winnard; Natalia Barros; Jose Gilberto Biondo Jr; Wilson de Figueiredo; Paul Robert Hering; Jaqueline Mourao; Ewerton Waki

IBM z/OS V2R2: User Interfaces
Keith Winnard; Paul Robert Hering

More Just Added >

Top Titles

Learning Spark
Holden Karau; Andy Konwinski; Patrick Wendell; Matei Zaharia

Hadoop: The Definitive Guide, 4th Edition
Tom White

Head First Java, 2nd Edition
Kathy Sierra; Bert Bates

Head First Design Patterns
Eric Freeman; Elisabeth Robson; Bert Bates; Kathy Sierra

Head First Android Development
Dawn Griffiths; David Griffiths

Advanced Analytics with Spark
Sandy Ryza; Uri Laserson; Sean Owen; Josh Wills

Titles by Popularity >

About Safari Books Online • Terms of Service • Privacy Policy • Contact Us • Help • Accessibility • Category Map • RSS
• Sign Out & Clear Session

Copyright 2015 Safari Books Online. All rights reserved.

Class Policies

All class announcements will be made through Canvas

- Check frequently, or sign up for email alerts

Class participation is expected — online participation is graded

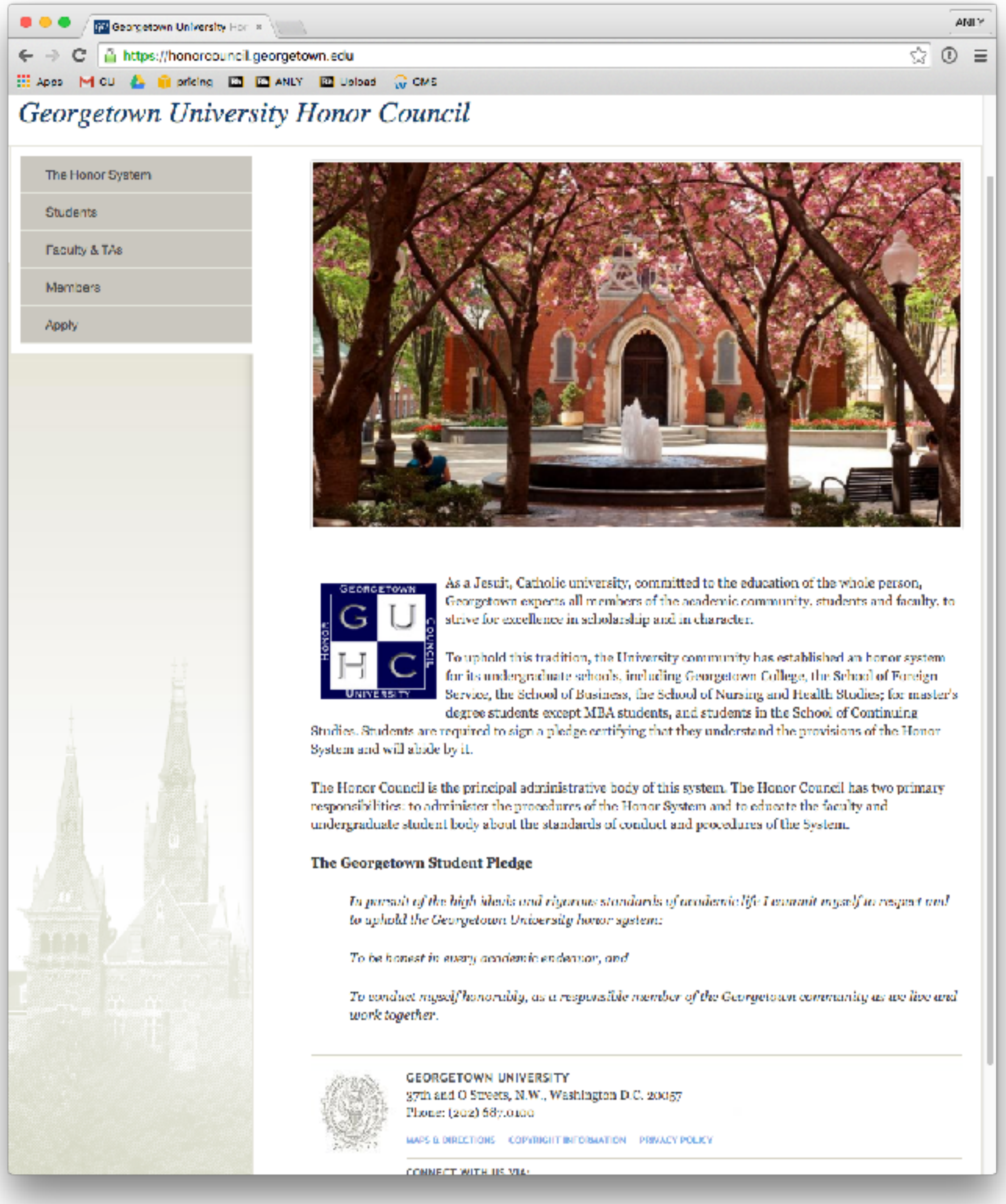
- Ask & answer questions
- Discuss relevant course events.

Assignments are due on Friday at 11:59pm

- Late homework will only be accepted in exceptional circumstances.
- Collaboration is allowed, but must be documented.
- You are expected to submit your own work.

Follow the Georgetown Student Pledge

- Please confirm that you will follow using Canvas.



Georgetown University Honor Council

As a Jesuit, Catholic university, committed to the education of the whole person, Georgetown expects all members of the academic community, students and faculty, to strive for excellence in scholarship and in character.

To uphold this tradition, the University community has established an honor system for its undergraduate schools, including Georgetown College, the School of Foreign Service, the School of Business, the School of Nursing and Health Studies; for master's degree students except MBA students, and students in the School of Continuing Studies. Students are required to sign a pledge certifying that they understand the provisions of the Honor System and will abide by it.

The Honor Council is the principal administrative body of this system. The Honor Council has two primary responsibilities: to administer the procedures of the Honor System and to educate the faculty and undergraduate student body about the standards of conduct and procedures of the System.

The Georgetown Student Pledge

> *In pursuit of the high ideals and rigorous standards of academic life I commit myself to respect and to uphold the Georgetown University honor system:*
>
> *To be honest in every academic endeavor, and*
>
> *To conduct myself honorably, as a responsible member of the Georgetown community as we live and work together.*

GEORGETOWN UNIVERSITY
37th and O Streets, N.W., Washington D.C. 20057
Phone: (202) 687.0100

MAPS & DIRECTIONS COPYRIGHT INFORMATION PRIVACY POLICY

CONNECT WITH US VIA:

Google Survey Results

http://bit.ly/ANLY502-2017-Responses

https://en.wikipedia.org/wiki/List_of_cloud_types#/media/File:Cirrus_clouds2.jpg

The Cloud…

The Cloud

Typical machine room

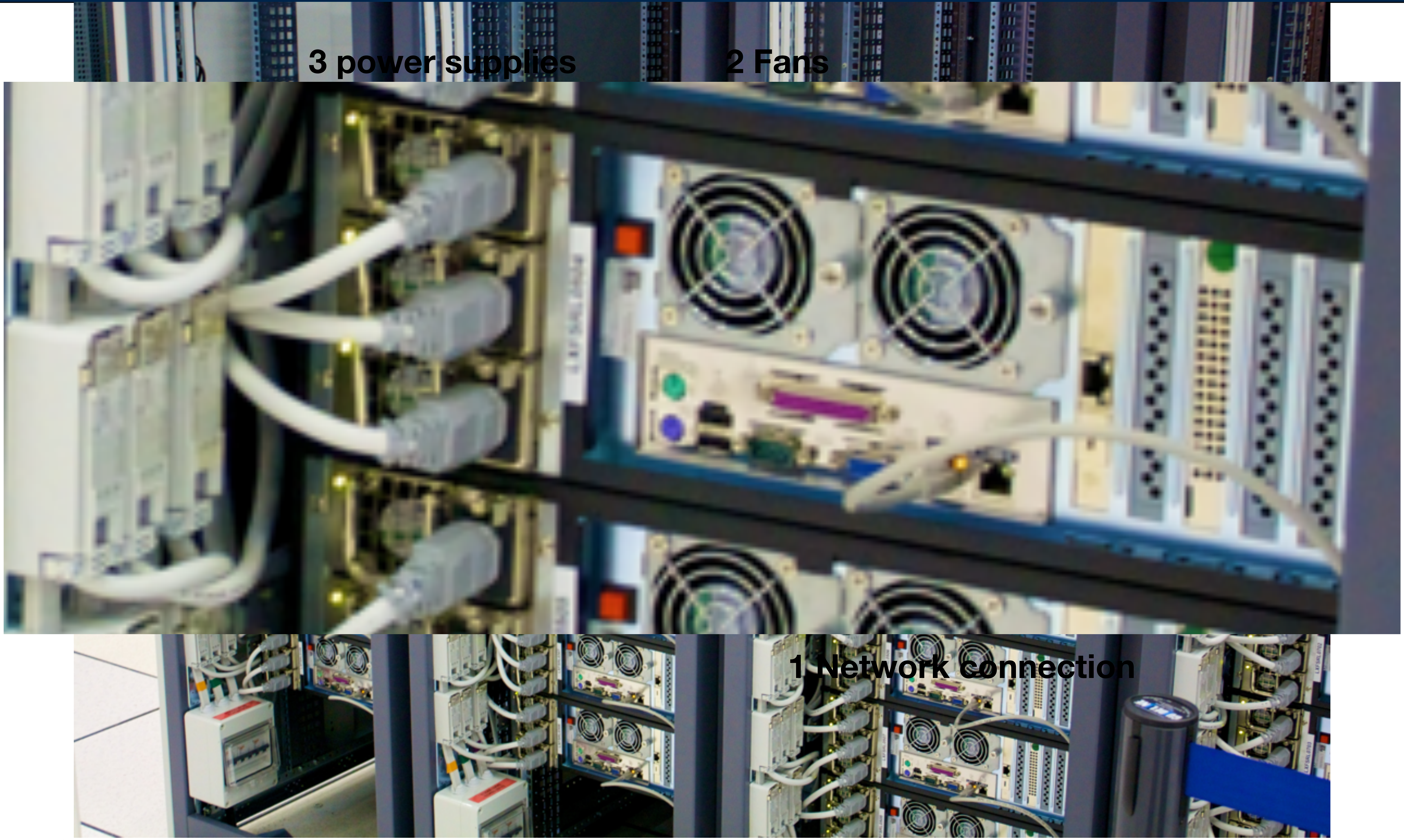
http://www.flickr.com/photos/torkildr/3462606643/sizes/l/in/photostream/

The Cloud

Many identical servers — each with disk, CPU and network

http://www.flickr.com/photos/torkildr/3462607995/in/photostream/

The Cloud



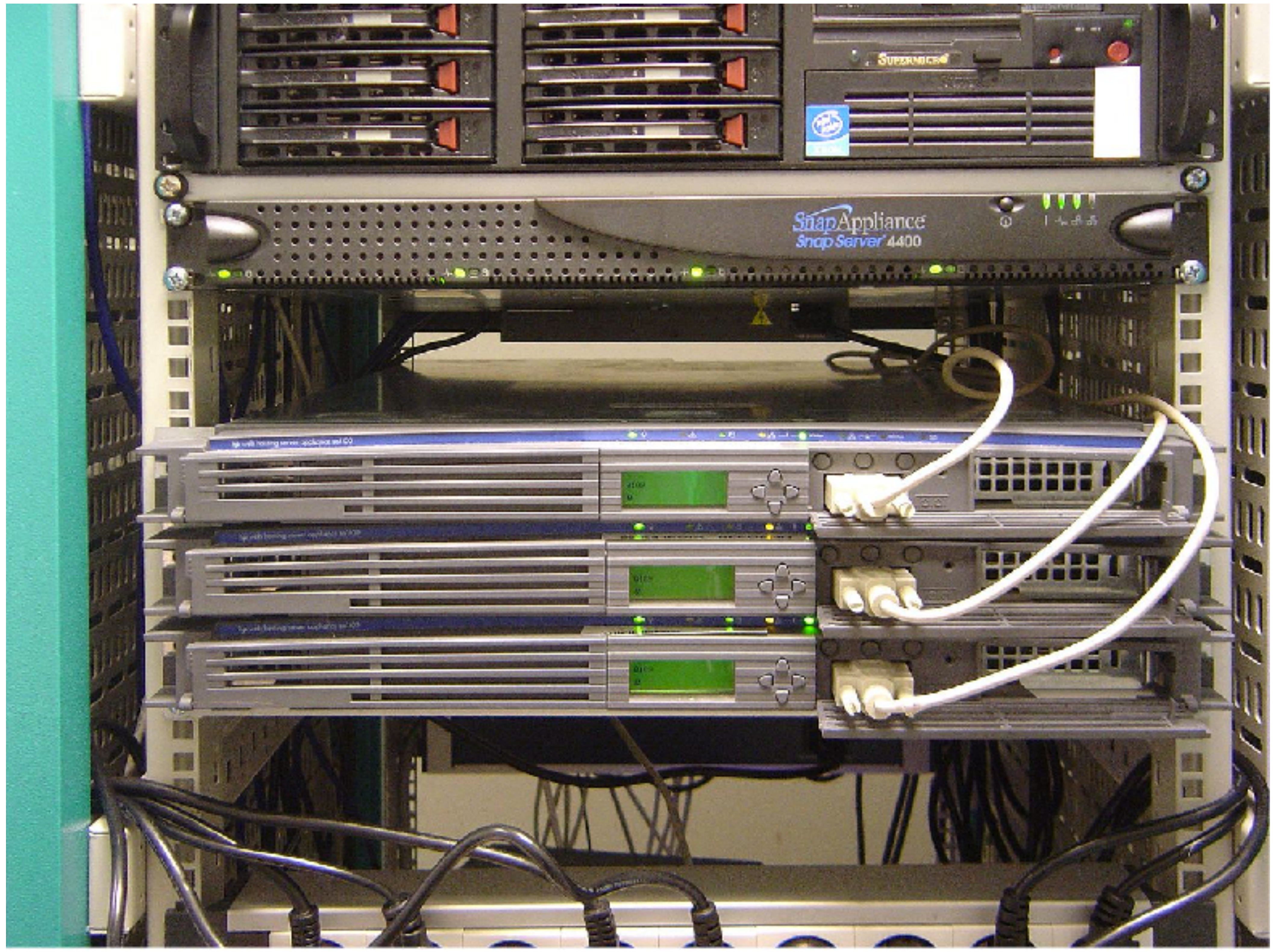
Systematic power distribution; 3 power supplies; 2 fans; 1 network connection

http://www.flickr.com/photos/torkildr/3463419826/

Early machine rooms: an equipment menagerie

This created significant manageability problems:

- Each machine had a distinct hardware and software configuration
- When a machine failed, it’s services couldn’t be readily moved to another.



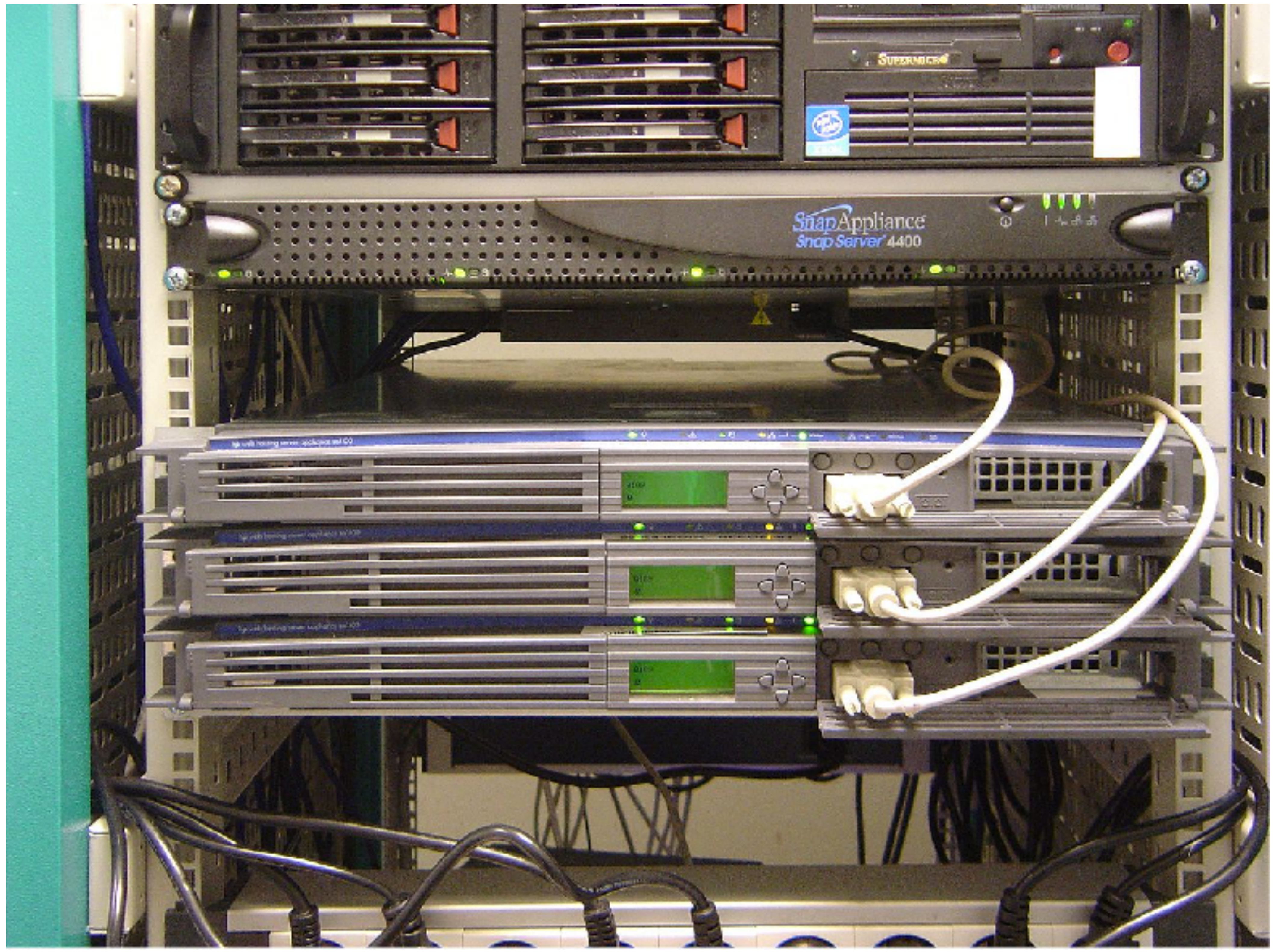
https://en.wikipedia.org/wiki/Data_center

1990s

Modern machine rooms have racks and racks of identical equipment.

Advantages:

- Consistent wiring plan.
- No computer is unique; easy fail-over and replacement



https://en.wikipedia.org/wiki/Data_center

1990s

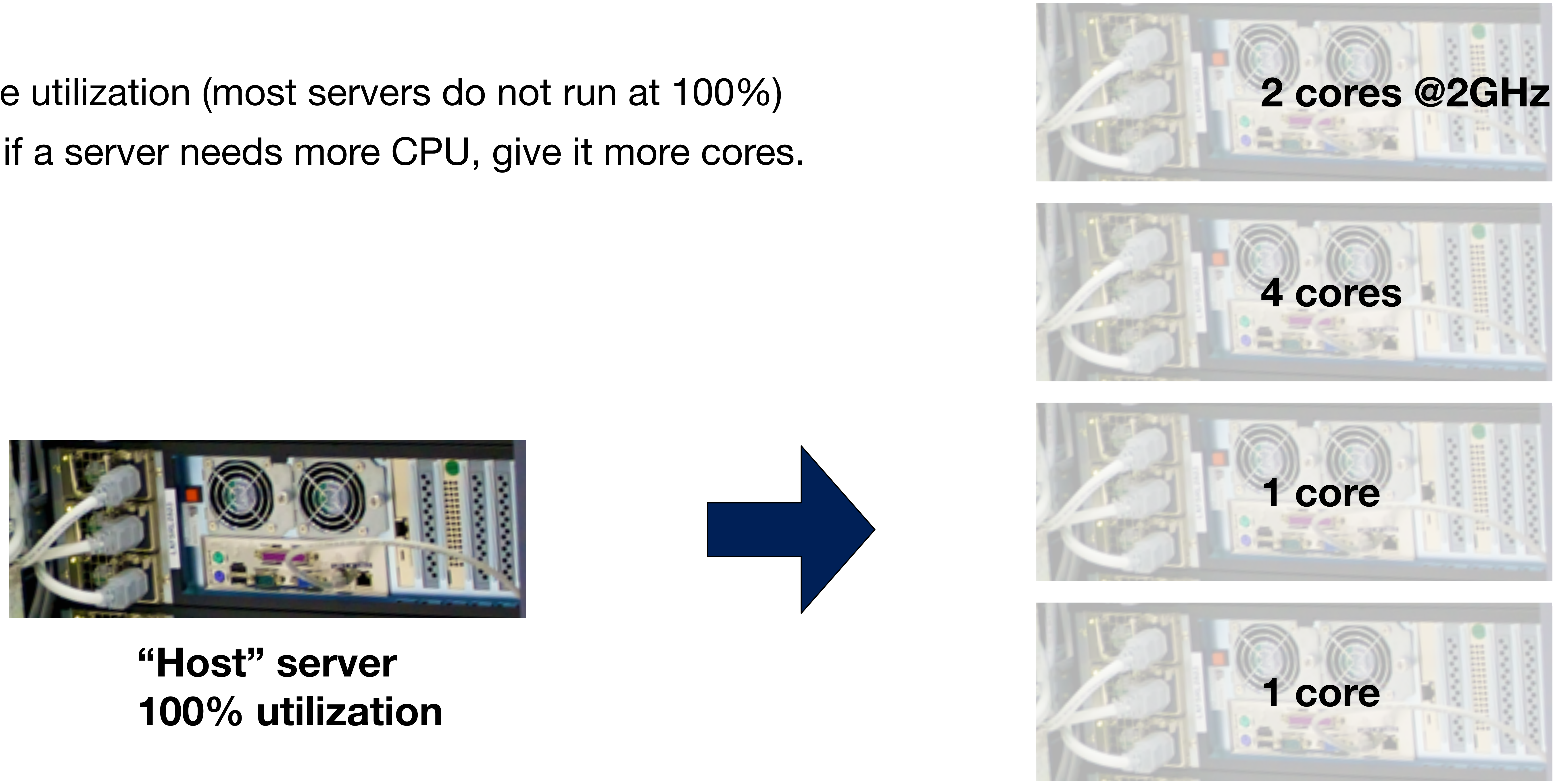
2005

Modern data centers virtualize servers, storage, and networks.

Server virtualization makes one server look like many

Advantages:

- Better hardware utilization (most servers do not run at 100%)
- Better scaling: if a server needs more CPU, give it more cores.



2 cores @2GHz

4 cores

1 core

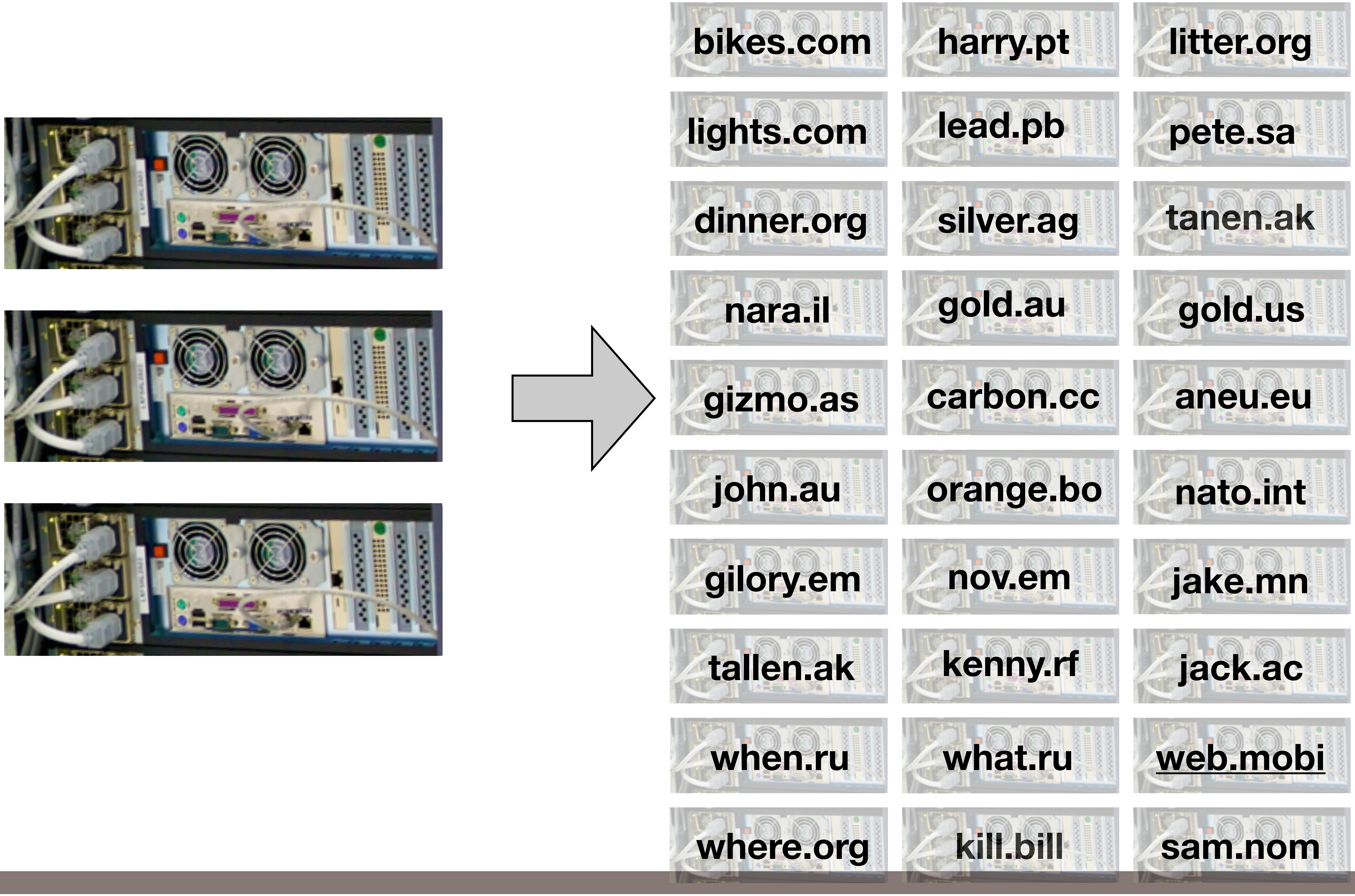
1 core

“Host” server
100% utilization

8 cores @ 2GHz

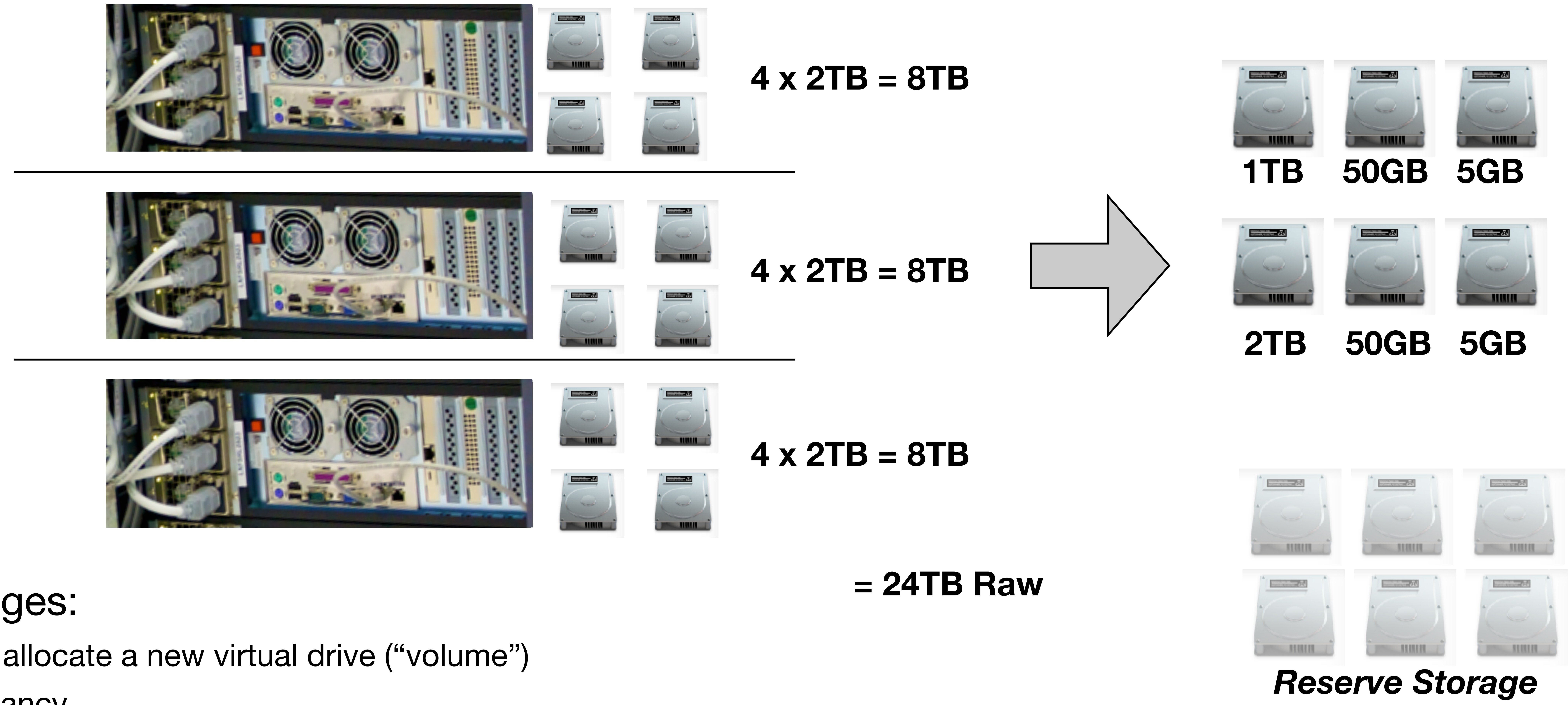
“Guest” operating systems
25% utilization each

This lets a few machines simulate many domains.



bikes.com	harry.pt	litter.org
lights.com	lead.pb	pete.sa
dinner.org	silver.ag	tanen.ak
nara.il	gold.au	gold.us
gizmo.as	carbon.cc	aneu.eu
john.au	orange.bo	nato.int
gilory.em	nov.em	jake.mn
tallen.ak	kenny.rf	jack.ac
when.ru	what.ru	web.mobi
where.org	kill.bill	sam.nom

Storage virtualization uses multiple servers to create the appearance of highly reliable storage arrays.



Advantages:

- Easy to allocate a new virtual drive (“volume”)
- Redundancy
- “Snapshot” and “Clone”
- Easy sharing of read-only volumes between servers.

Redundancy protects data against drive failure.

Every drive will eventually fail.

RAID — Redundant Array of Inexpensive Drives

- Stores data and “parity bits” across drives.
- Typical overhead: 16%-50% — 3-6 drives in a *RAID Set.*
  - *RAID5 — Can tolerate the failure of 1 drive.*
  - *RAID6 — Can tolerate the failure of 2 drives.*
- Requires “rebuild” when a drive is replaced.
  - *Drives frequently fail during rebuild.*
- Hard to add more storage

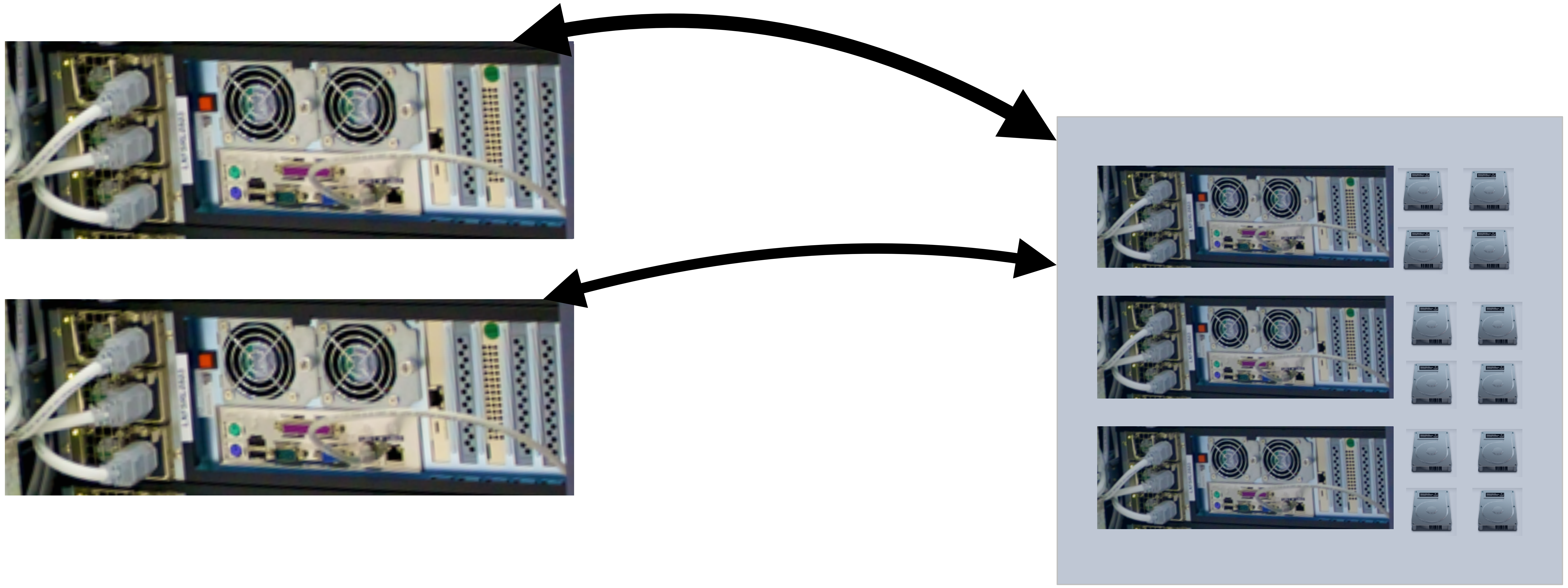
Store:

Replication

- Stores multiple copies of data on different drives.
- One copy gives protection against drive failures.
- Multiple copies gives increased performance
  - *can read from multiple drives at once.*
- Typical overhead: 200% - 300% (or more)
- Requires object copying when a drive fails.
- Easy to add more storage

Virtualized storage appears as a remote “file server”

Virtual drives are accessed over the network.



Traditional file server protocols include:

- iSCSI — Block read/write protocol. (rw for single computer, ro for multiple)
- NFS — File read/write protocol. (rw or ro for multiple computers.)

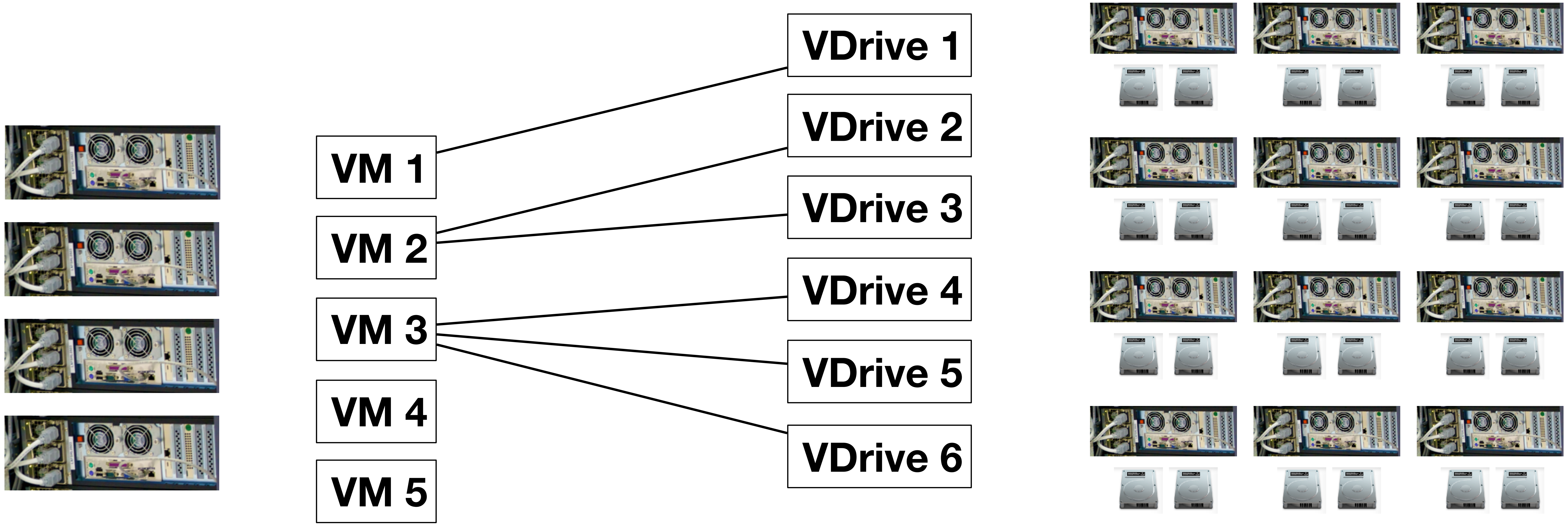
Big providers combine CPU and storage virtualization. Each virtual server runs on top of virtual storage.

Storage array holds:

- VM configurations
- VM drives.

If a drive fails, the array provides data availability

If a compute server fails, the VM restarts on another physical machine.



There are two primary ways that clusters are organized

“High-Performance” or “Scientific” Computing

High performance CPUs

High performance storage

Supercomputer K
Fastest in the world
8.162 petaflops
Hundreds of identical nodes:

Scientific Node
Fastest per-CPU speed

Data-Centric Computing

Commodity CPUs

Integrated Storage

Data-Centric Node
Cheapest $ per compute
Cheapest $ per storage

Scientific computing: fastest speed at any price

Systems have:

- Separate CPU and storage
- Optimized for floating point.

Separate control and data:

- TCP/IP for control over 10gig
- Fibre channel SAN for disks
- Distributed file system, lets code quickly fetch data from high-performance disk servers.

Design goals:

- High component reliability = no failures.
- Predictable job execution.

Supercomputer "K computer" Takes First Place in World

Achieves world's best performance of 8.162 petaflops to lead TOP500 list

RIKEN,Fujitsu Limited

Tokyo, June 20, 2011

High performance CPUs

High performance storage

672 racks = 68,544 CPUS
8.162 petaflops

Data-centric (commercial) computing: Cheapest cost / biggest data

Systems have:

- CPU and storage in the same box

Integrated control and data:

- Data are distributed in different machines.
- Code goes to data to run.
- Results stored locally or sent to other nodes.

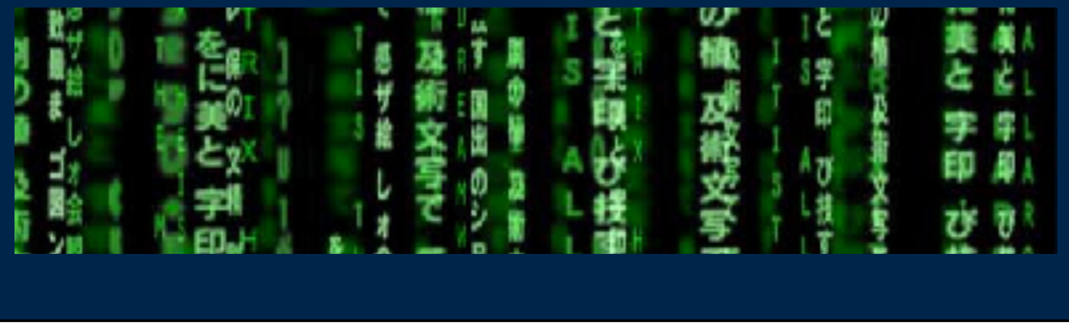
Design goals:

- Failure tolerant
  - *—If a computer fails once every 4 years, 1200 computers may average a failure every day!*
- Commodity hardware
  - *—Chapter to buy 4 computers with 16 cores thank 1 computer with 64 cores*

Commodity hardware

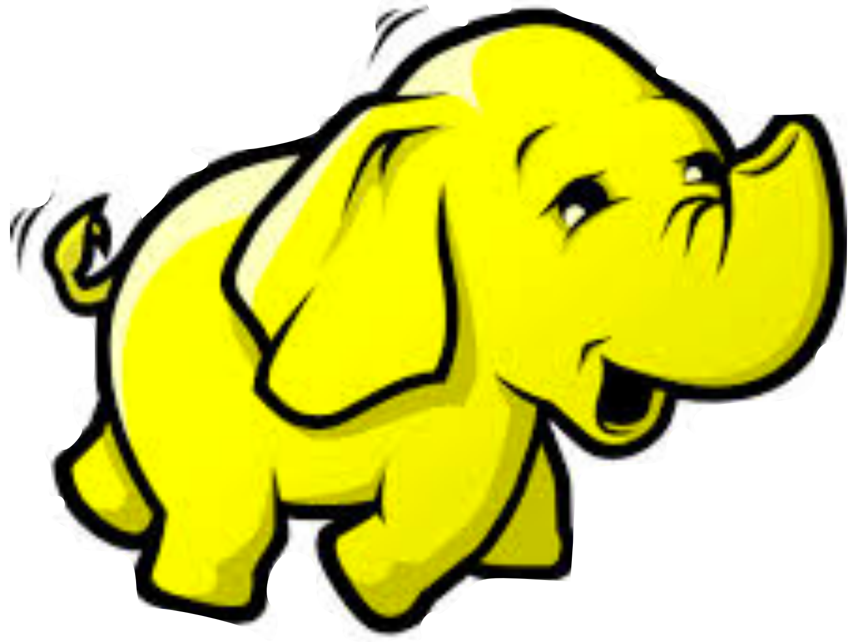
Massive Data Technology (for ANLY 502)

Massive Data Technology:
Specific technology that we use in this course.



Program Layer — code that you write to manipulate the data

- **Python**, Scala, Java, etc.



Software Infrastructure Layer — where your code runs

- YARN, Hadoop, MapReduce, Pig, Spark, etc.
- Databases: HBase, Hive, Impala

Operating System Layer — what you log into

- **Linux (Centos)**, Windows

Virtualization Layer — the runtime environment

- **Xen**

Hardware Layer — the physical hardware on which the VMs run

- Doesn't matter what it is!

In 2011, the National Institute of Standards and Technology defined a standard terminology for cloud computing.

Special Publication 800-145: The NIST Definition of Cloud Computing

Essential Characteristics:

- On-demand self-service ✔
- Broad network access ✔
- Resource pooling ✔
- Rapid elasticity ✔
- Measures service ✔

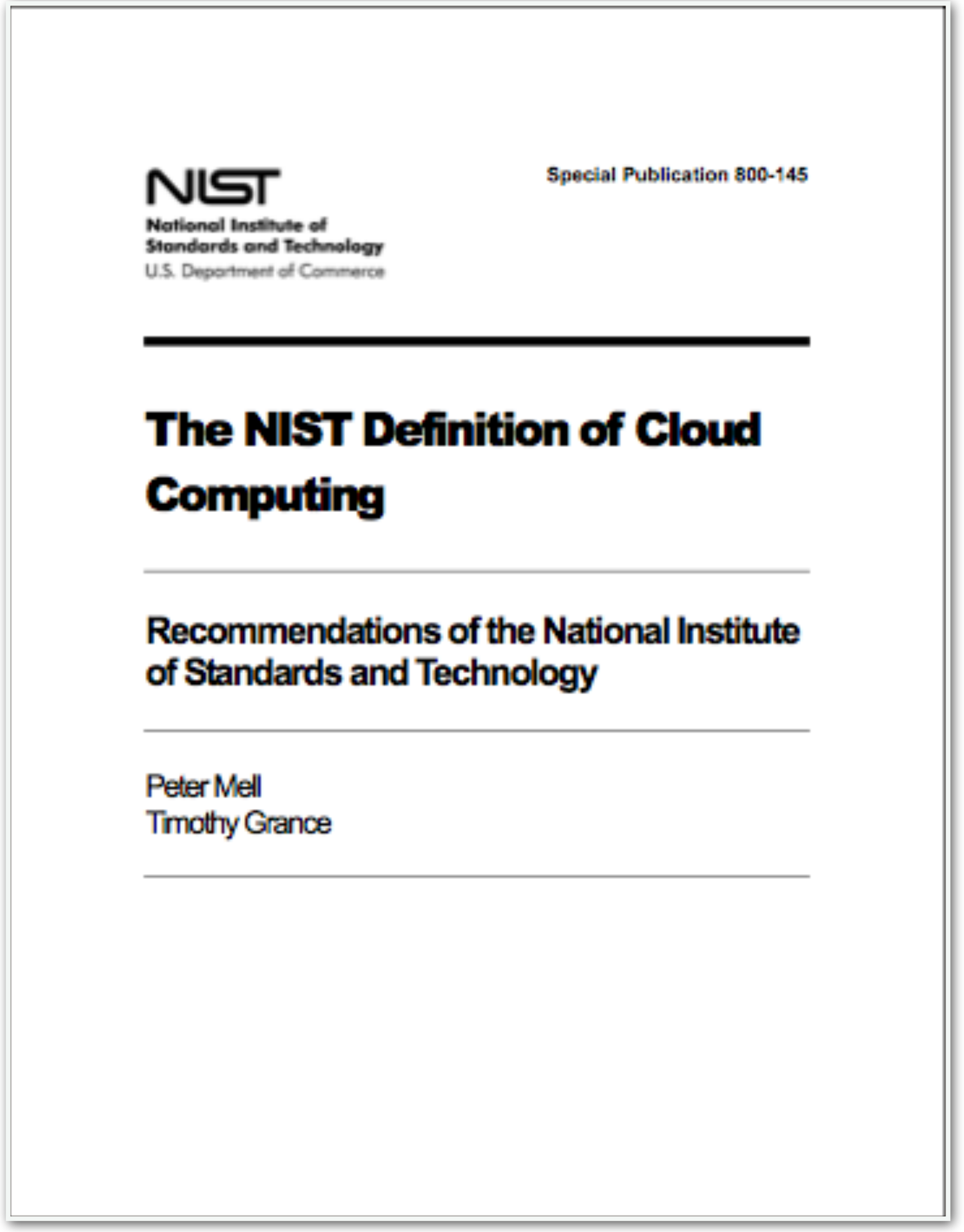
Deployment Models:

- Private cloud ✔
- Community cloud
- Public cloud ✔
- Hybrid cloud

Service Models:

- Software as a Service (SaaS) ✔
- Platform as a Service (PaaS) ✔
- Infrastructure as a Service (IaaS) ✔

Amazon offers most of these models!



NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Special Publication 800-145

The NIST Definition of Cloud Computing

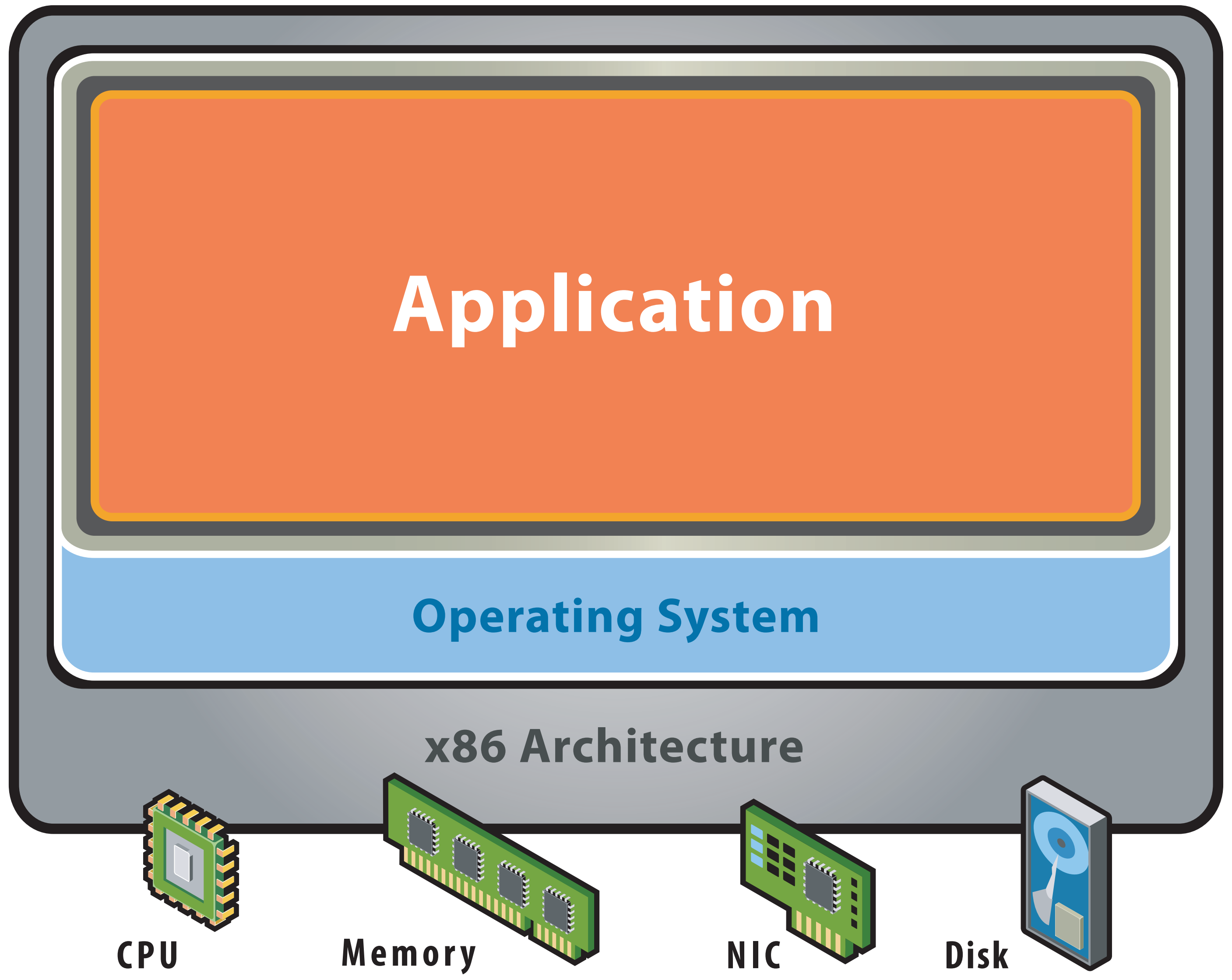
Recommendations of the National Institute of Standards and Technology

Peter Mell
Timothy Grance

Labs & assignments will be on Amazon Web Services.

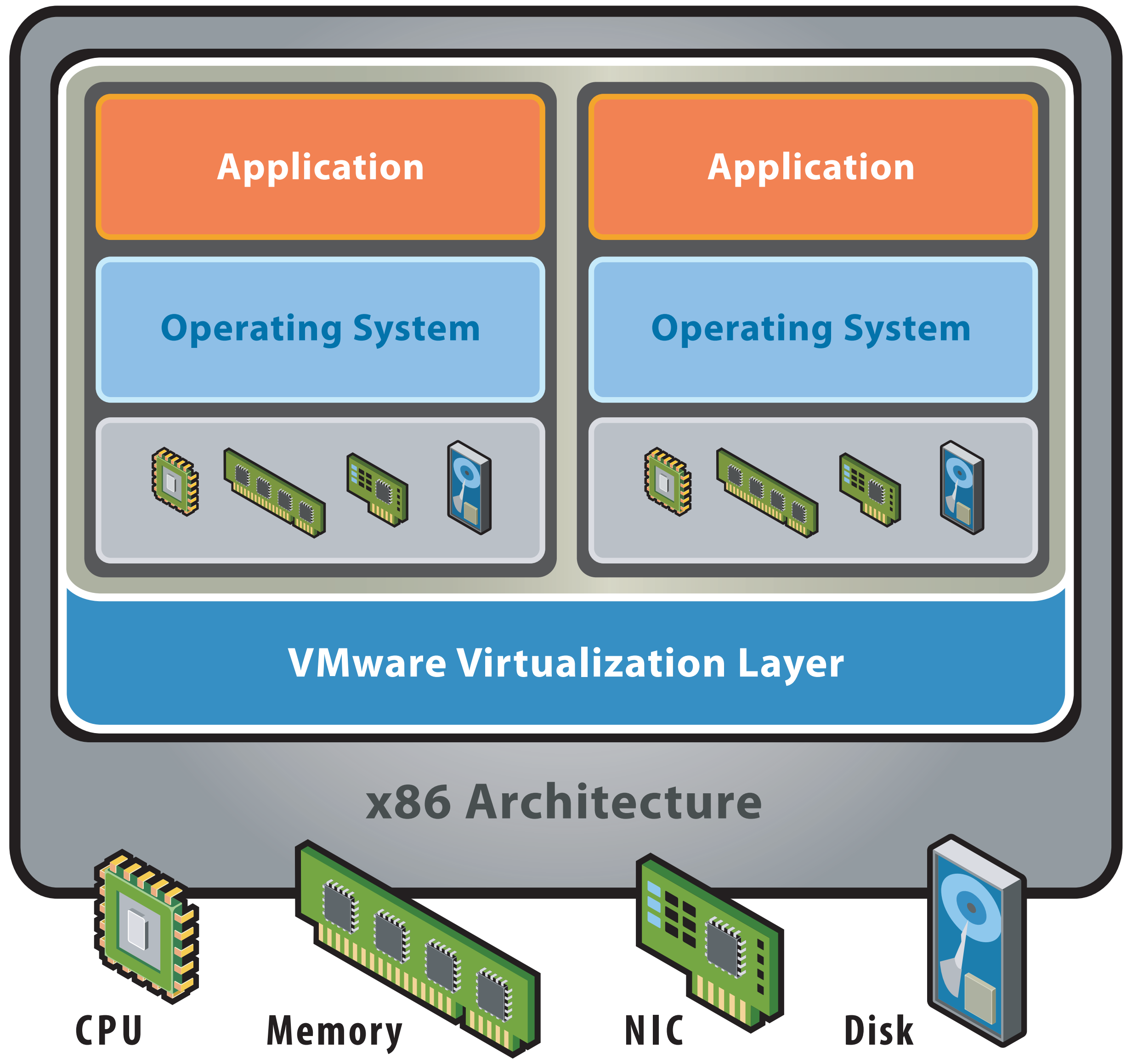
*http://theatln.tc/1GVLpOM

Virtualization Basic Idea: A computer within a computer



Before Virtualization:

- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure



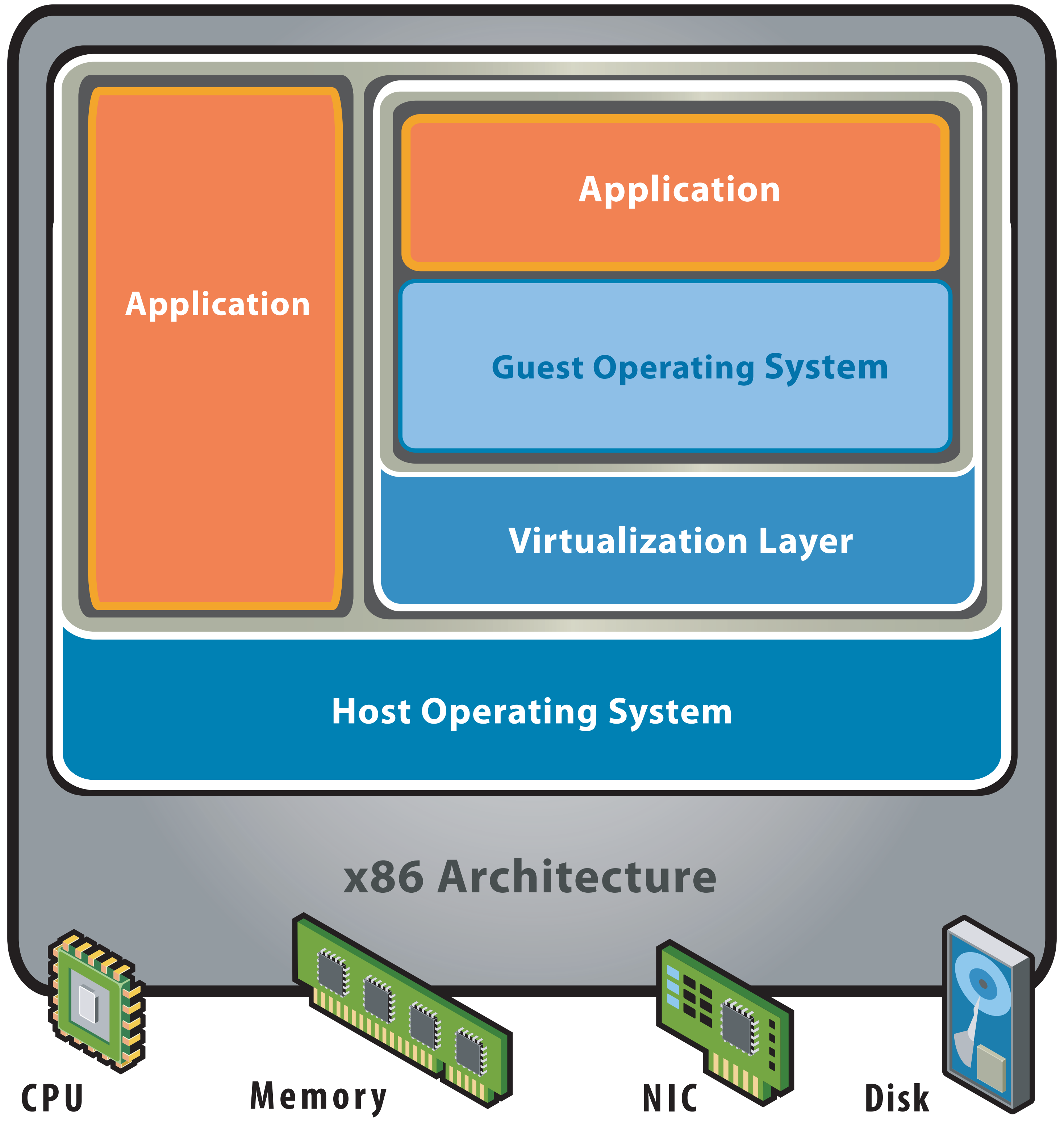
After Virtualization:

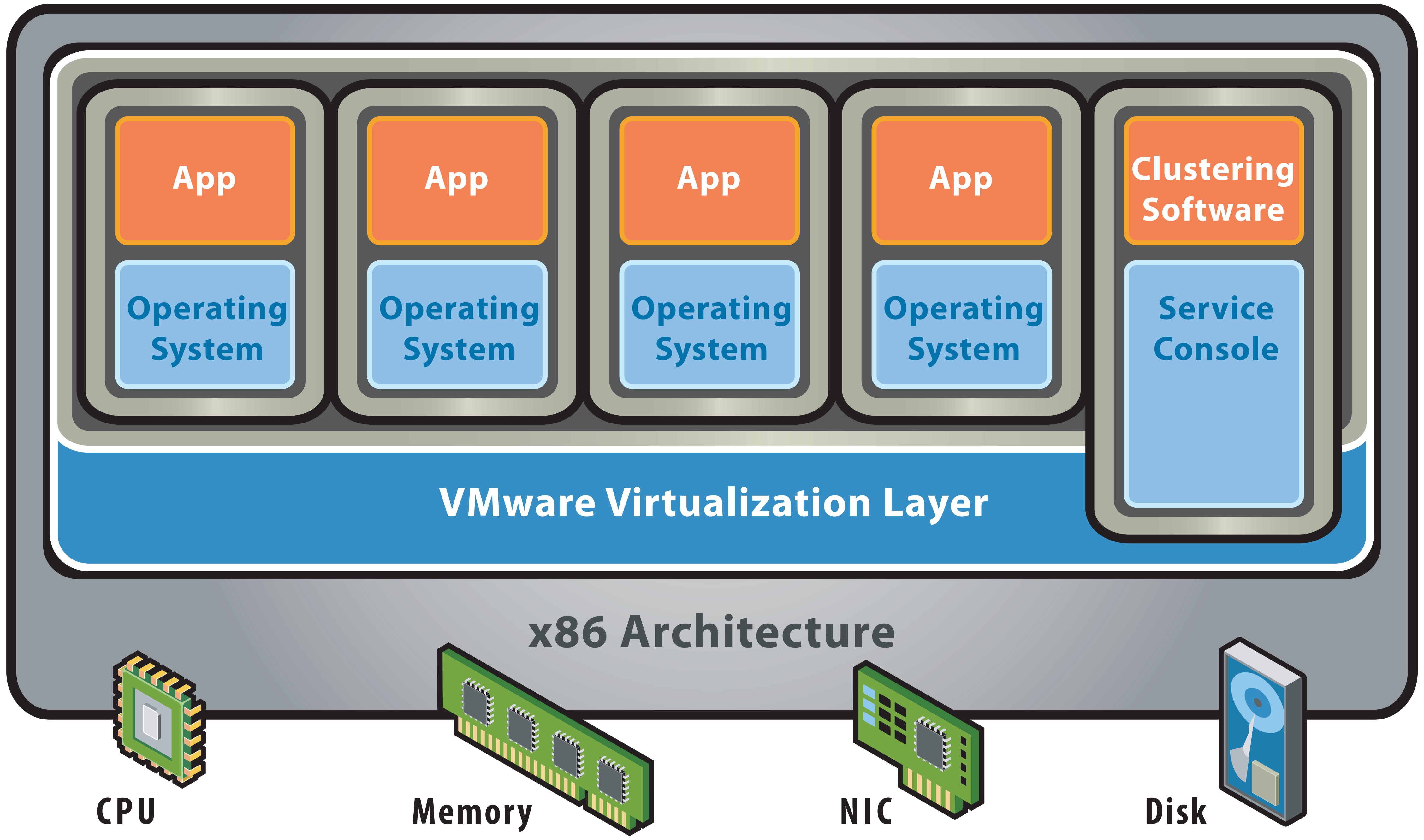
- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines

Virtualization Overview

- https://www.vmware.com/pdf/virtualization.pdf

Two virtualization architectures: “Hosted” and “Bare-Metal”





Hosted Architecture

- Installs and runs as an application
- Relies on host OS for device support and physical resource management

Bare-Metal (Hypervisor) Architecture

- Lean virtualization-centric kernel
- Service Console for agents and helper applications

Why Amazon — Don't they sell books?

July 5, 1994 — Amazon.com was founded by Jeff Bezos

- (Originally named "Cadabra")
- Renamed "Amazon" in 1995 with goal of being the "biggest" store in the world.
- First book ordered in 1995, *Fluid Concepts and Creative Analogies.**

In 1998, showing a single web page on Amazon.com required more than 100 computers!

- Amazon made organizing thousands of computers an institutional priority.

In 2006, Amazon started making its systems available as a commodity

- Simple Queue Service (SQS) — Reliable messages up to 256KB in size.
- Elastic Compute Cloud (EC2) — virtual machines
- Simple Storage Service (S3) — unlimited storage

Amazon Elastic
MapReduce

*http://theatln.tc/1GVLpOM

Pictures of our physical data center are not very useful…

http://bit.ly/amazon-data-center1

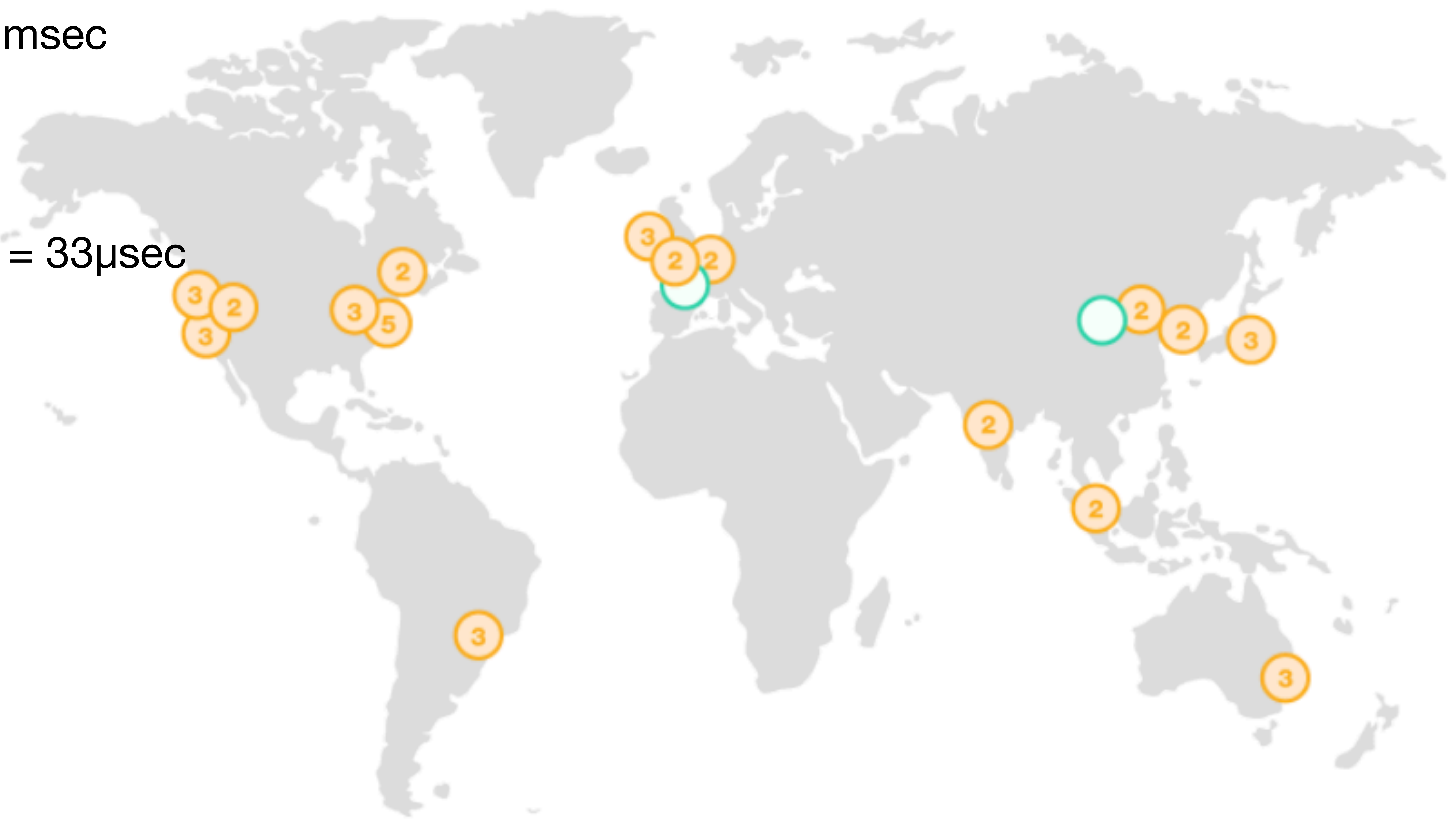
More important: Where the data centers are.

Location matters:

- Speed of light: 300,000 Km/sec = 300Mm/s
- Distance to Seattle: ≈5,000 Km = 5Mm
- Minimum time to Seattle: 5 Mm ÷ 300 Mm/s = 1.6 msec

- Distance to Reston: ≈ 10Km
- Minimum time to Reston: 10 Km ÷ 300,000 Km/sec = 33μsec

Global Infrastructure



● # **Region & Number of Availability Zones**

AWS GovCloud (2)

US West
Oregon (3), Northern California (3)

US East
Northern Virginia (5), Ohio (3)

Canada
Central (2)

South America
São Paulo (3)

Europe
Ireland (3), Frankfurt (2), London (2)

Asia Pacific
Singapore (2), Sydney (3), Tokyo (3), Seoul (2), Mumbai (2)

China
Beijing (2)

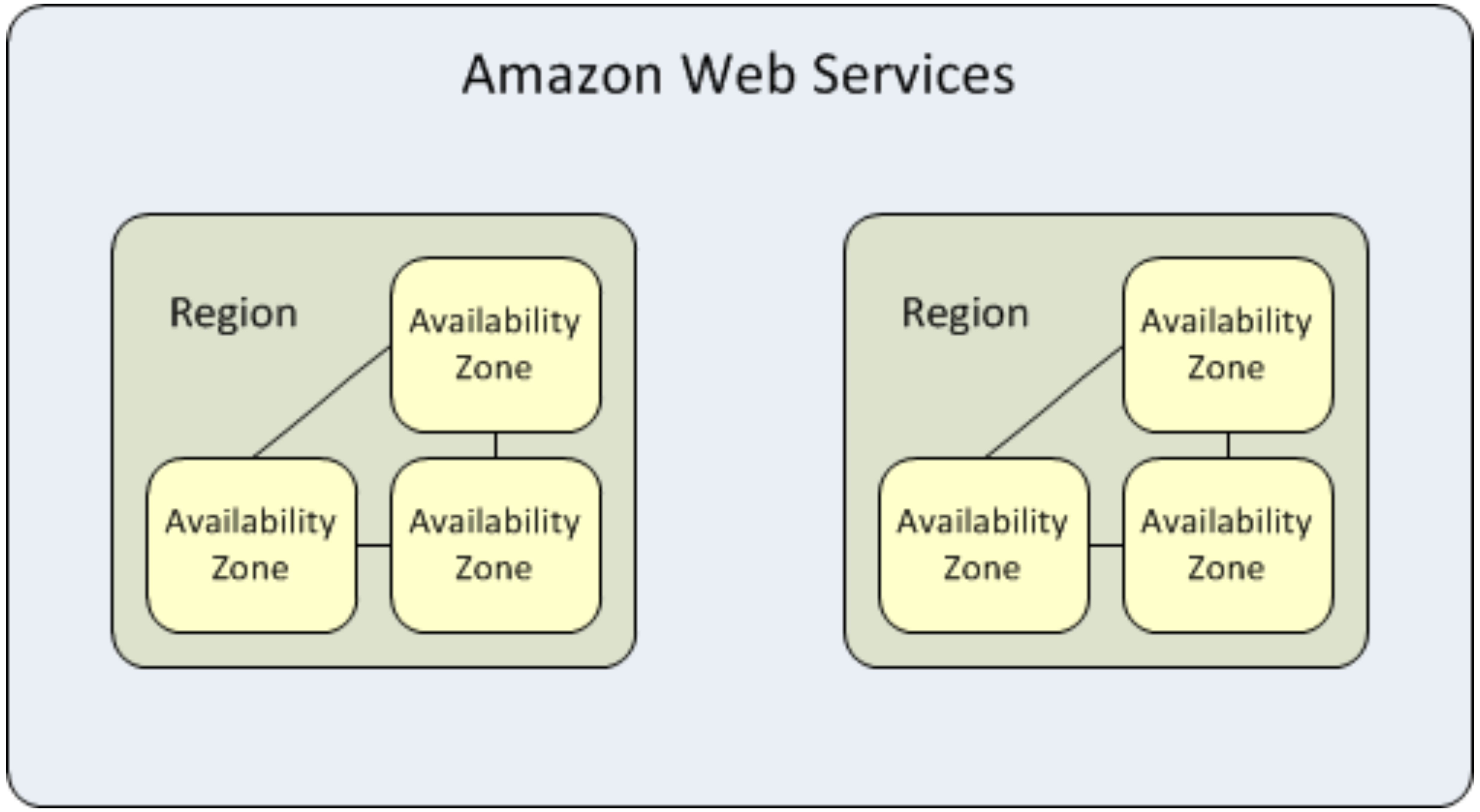
○ **New Regi**

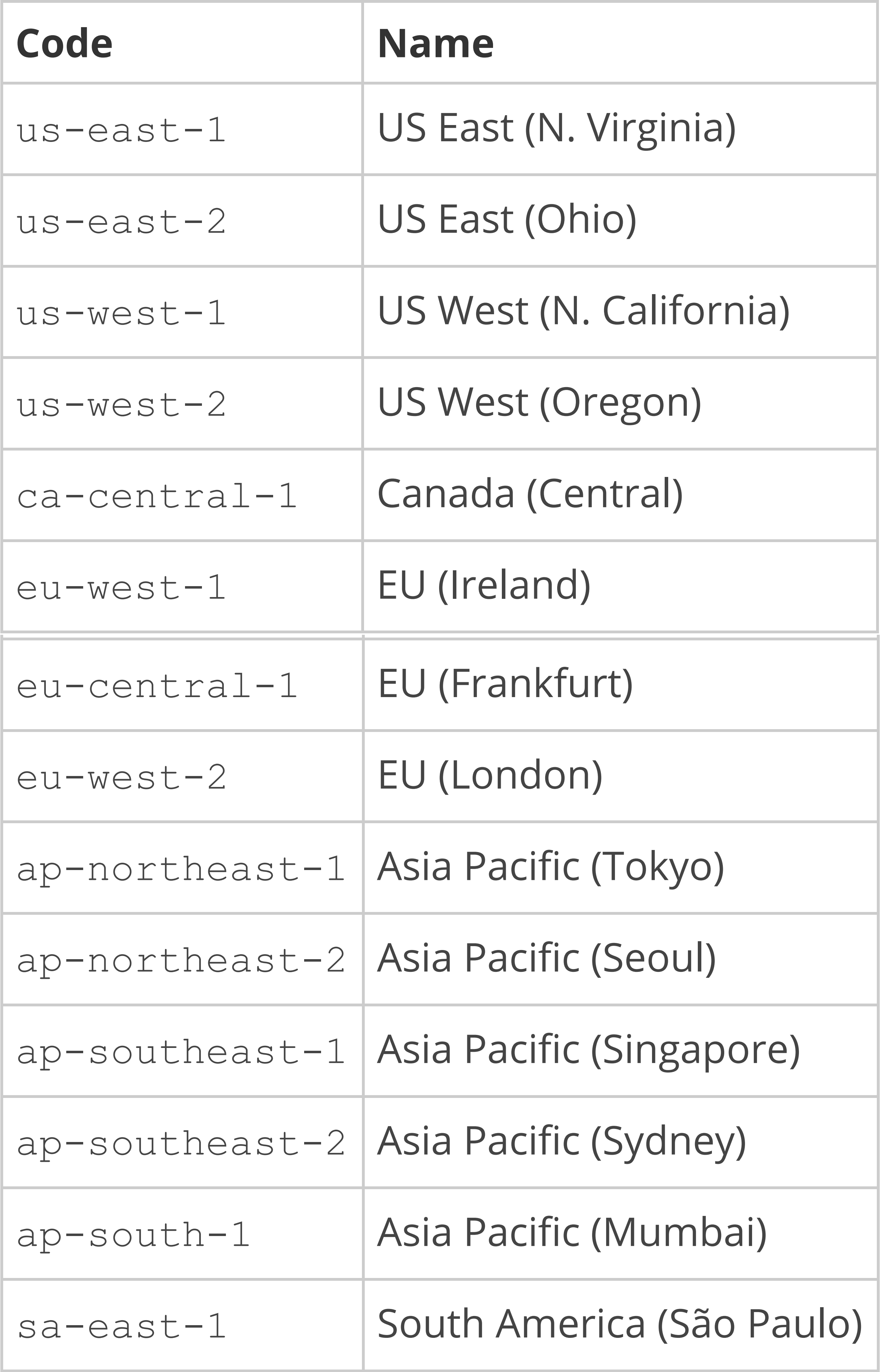
Paris

Ningxia

https://aws.amazon.com/about-aws/global-infrastructure/

AWS is divided into “regions” and “availability zones”





Code	Name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
ca-central-1	Canada (Central)
eu-west-1	EU (Ireland)
eu-central-1	EU (Frankfurt)
eu-west-2	EU (London)
ap-northeast-1	Asia Pacific (Tokyo)
ap-northeast-2	Asia Pacific (Seoul)
ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
ap-south-1	Asia Pacific (Mumbai)
sa-east-1	South America (São Paulo)

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

Alternatives to Amazon

Top Tier — scalable VMs, services, etc.:

- Google Cloud Platform — https://cloud.google.com
- Microsoft Azure — https://azure.microsoft.com/en-us/
- Rackspace — http://www.rackspace.com

Bargain basement:

- Dreamhost — http://www.dreamhost.com/
- WebFaction — https://www.webfaction.com

Services may charge for:

- Computation — Virtual Machines
- Storage
- Bandwidth
- Special APIs and Services
- Setup

We use Amazon:

- Currently best developed of the services
- Excellent documentation
- Many online tutorials

Amazon has “first mover advantage” and has not [yet] slipped!

Amazon Web Services

- EC2
- CloudWatch
- EBS
- EMR

AWS Educate

Amazon’s “educate” program entitles you to $100 grant.

- http://aws.amazon.com/education/awseducate/
- In addition to “free tier.”

Benefits to students:

- $100/student — at member institutions (GU is a member institution)
- AWS Training — Free access to labs
- Curated Content — Free access to AWS content for homework, labs, and self-study
- Collaboration Tools — Student portal access, virtual events, provide feedback.

Use “free tier” to develop your code.

Use “educate” to run EMR and ES jobs.

Each student will have $100 of "free" Amazon time.

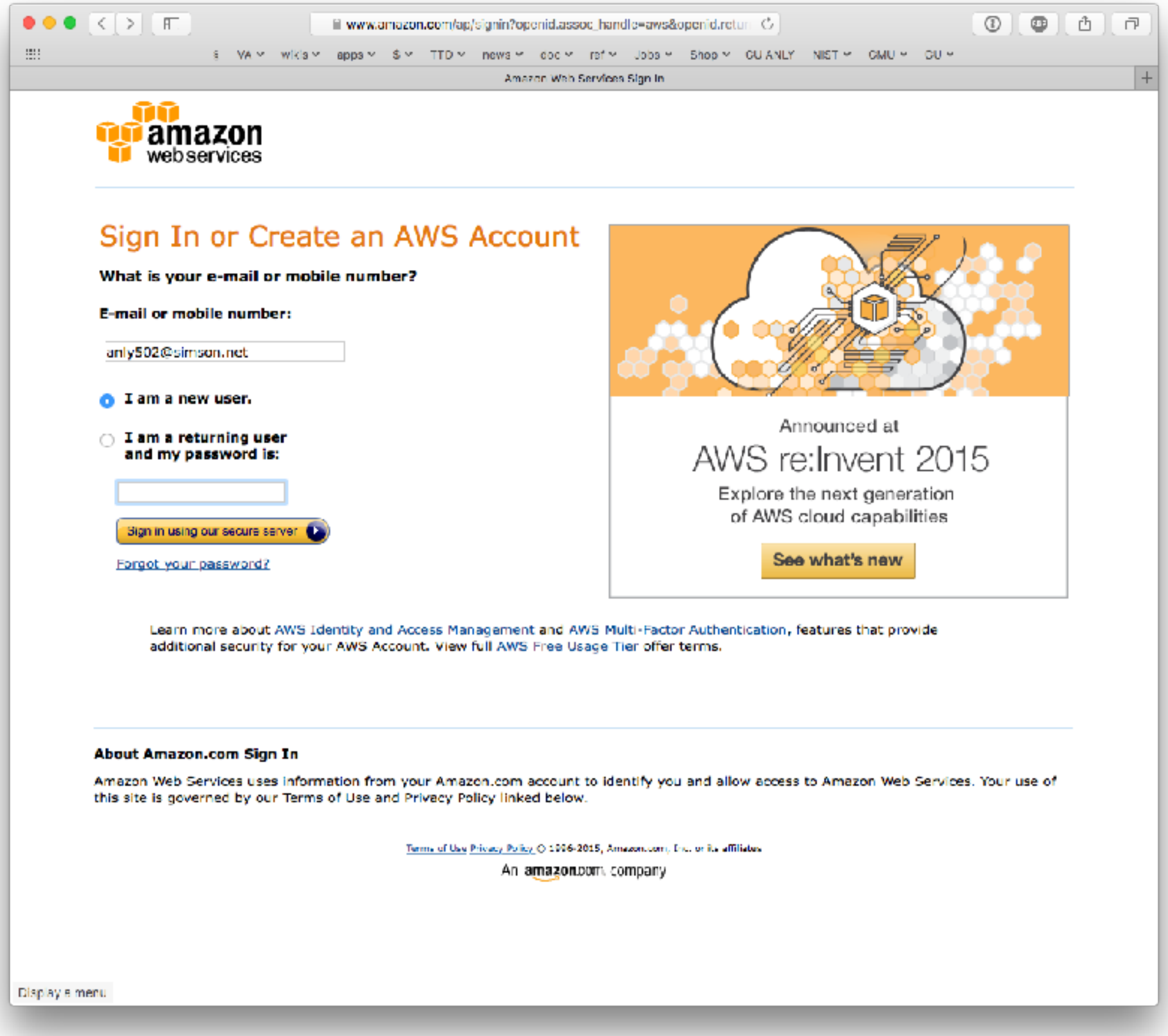
You can do a lot with $100:

Price for a General Purpose t2.medium (2 CPU, 4 GB, Variable ECU, EBS) [1]	$.052/hour ($8.74/week)
Price for m3.2xlarge Elastic Map Reduce (4 CPU, 26 ECU, 30GB, 2x80 SSD) [2]	$0.532/hour (EC2) + $0.140/hour (EMR) = $0.672/hour ($112.90/week)
EBS General Purpose Storage (SSD) [3]	$0.10/GB-month ($2.25 to store 100GB for a week)
EBS Magnetic volumes	$0.05/GB-month + $0.05 per 1 million I/O requests
EBS "snapshots"	$0.95/GB-month
Price to access public EBS datasets from EC2	FREE

[1] https://aws.amazon.com/ec2/pricing/
[2] https://aws.amazon.com/elasticmapreduce/pricing/
[3] https://aws.amazon.com/ebs/pricing/

You need an account on Amazon Web Services (AWS)

1. Create an amazon.com account (if you don't have one already)
2. Go to aws.amazon.com and sign up for Amazon Web Services



www.amazon.com/ap/signin?openid.assoc_handle=aws&openid.retur
VA wikis apps $ TTD news doc ref Jobs Shop GU ANLY NIST CMU GU
Amazon Web Services Sign In
amazon web services

Sign In or Create an AWS Account

What is your e-mail or mobile number?

E-mail or mobile number: anly502@simson.net

● I am a new user.

○ I am a returning user and my password is:

Sign in using our secure server
Forgot your password?

Announced at

AWS re:Invent 2015

Explore the next generation of AWS cloud capabilities

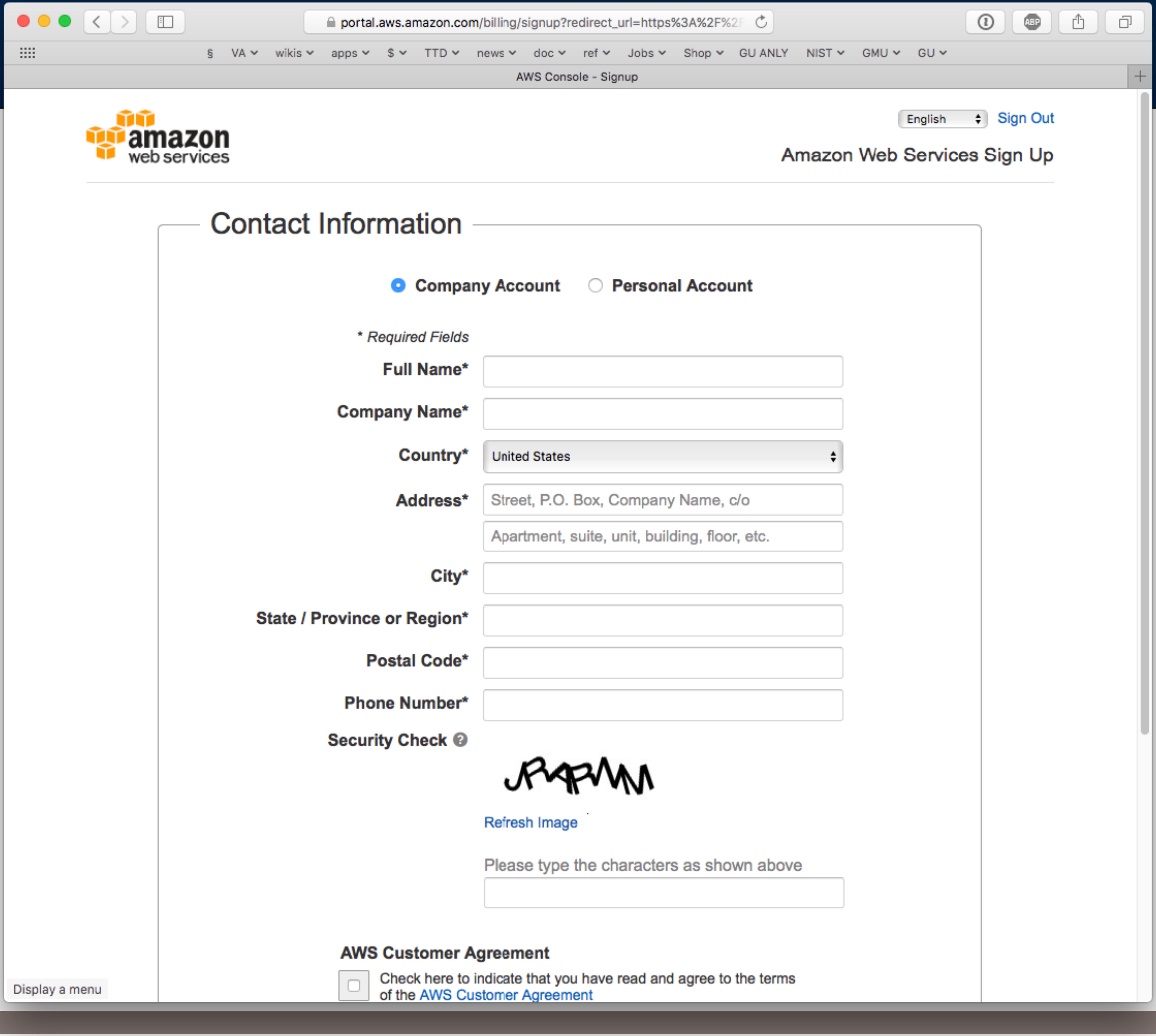
Learn more about AWS Identity and Access Management and AWS Multi-Factor Authentication, features that provide additional security for your AWS Account. View full AWS Free Usage Tier offer terms.

About Amazon.com Sign In

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services. Your use of this site is governed by our Terms of Use and Privacy Policy linked below.

Terms of Use Privacy Policy © 1996-2015, Amazon.com, Inc. or its affiliates

An amazon.com company



portal.aws.amazon.com/billing/signup?redirect_url=https%3A%2F%2F
§ VA wikis apps $ TTD news doc ref Jobs Shop GU ANLY NIST GMU GU
AWS Console - Signup
amazon web services
English
Display a menu

Sign Out

Amazon Web Services Sign Up

Contact Information

● Company Account ○ Personal Account

* Required Fields

Full Name*	
Company Name*	
Country*	United States
Address*	Street, P.O. Box, Company Name, c/o Apartment, suite, unit, building, floor, etc.
City*	
State / Province or Region*	
Postal Code*	
Phone Number*	
Security Check	JRRMN

Refresh Image

Please type the characters as shown above

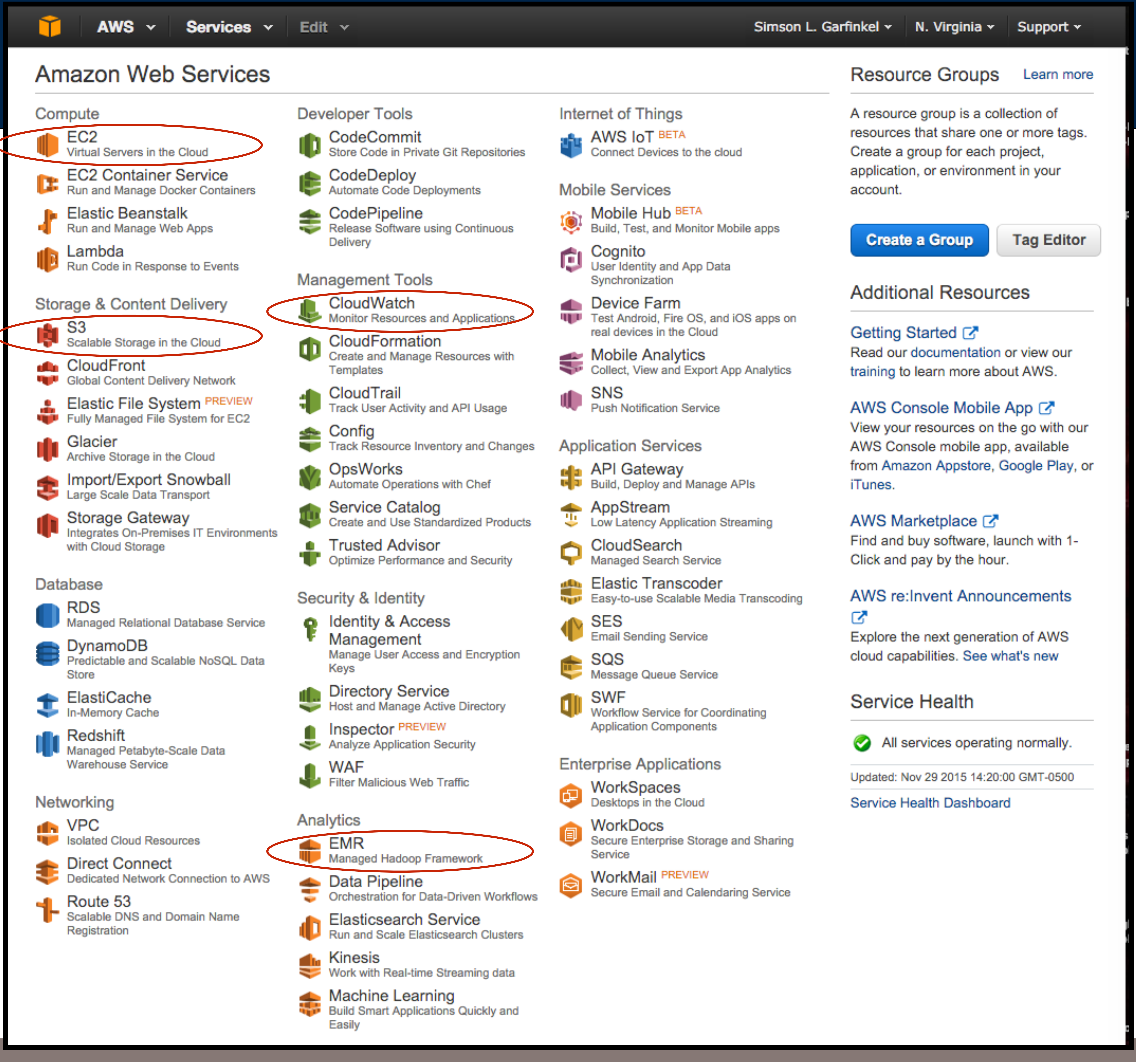
AWS Customer Agreement

□ Check here to indicate that you have read and agree to the terms of the AWS Customer Agreement

AWS has a lot of services:

In this course, we will focus on:

- EC2 — Virtual Servers
- CloudWatch
- S3 — Object-based Storage
- EMR — MapReduce & Spark



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
Integrates On-Premises IT Environments with Cloud Storage

Database

RDS
Managed Relational Database Service

DynamoDB
Predictable and Scalable NoSQL Data Store

ElastiCache
In-Memory Cache

Redshift
Managed Petabyte-Scale Data Warehouse Service

Networking

VPC
Isolated Cloud Resources

Direct Connect
Dedicated Network Connection to AWS

Route 53
Scalable DNS and Domain Name Registration

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

Analytics

EMR
Managed Hadoop Framework

Data Pipeline
Orchestration for Data-Driven Workflows

Elasticsearch Service
Run and Scale Elasticsearch Clusters

Kinesis
Work with Real-time Streaming data

Machine Learning
Build Smart Applications Quickly and Easily

Internet of Things

AWS IoT BETA
Connect Devices to the cloud

Mobile Services

Mobile Hub BETA
Build, Test, and Monitor Mobile apps

Cognito
User Identity and App Data Synchronization

Device Farm
Test Android, Fire OS, 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 Service

SQS
Message Queue Service

SWF
Workflow Service for Coordinating Application Components

Enterprise Applications

WorkSpaces
Desktops in the Cloud

WorkDocs
Secure Enterprise Storage and Sharing Service

WorkMail PREVIEW
Secure Email and Calendaring Service

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.

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

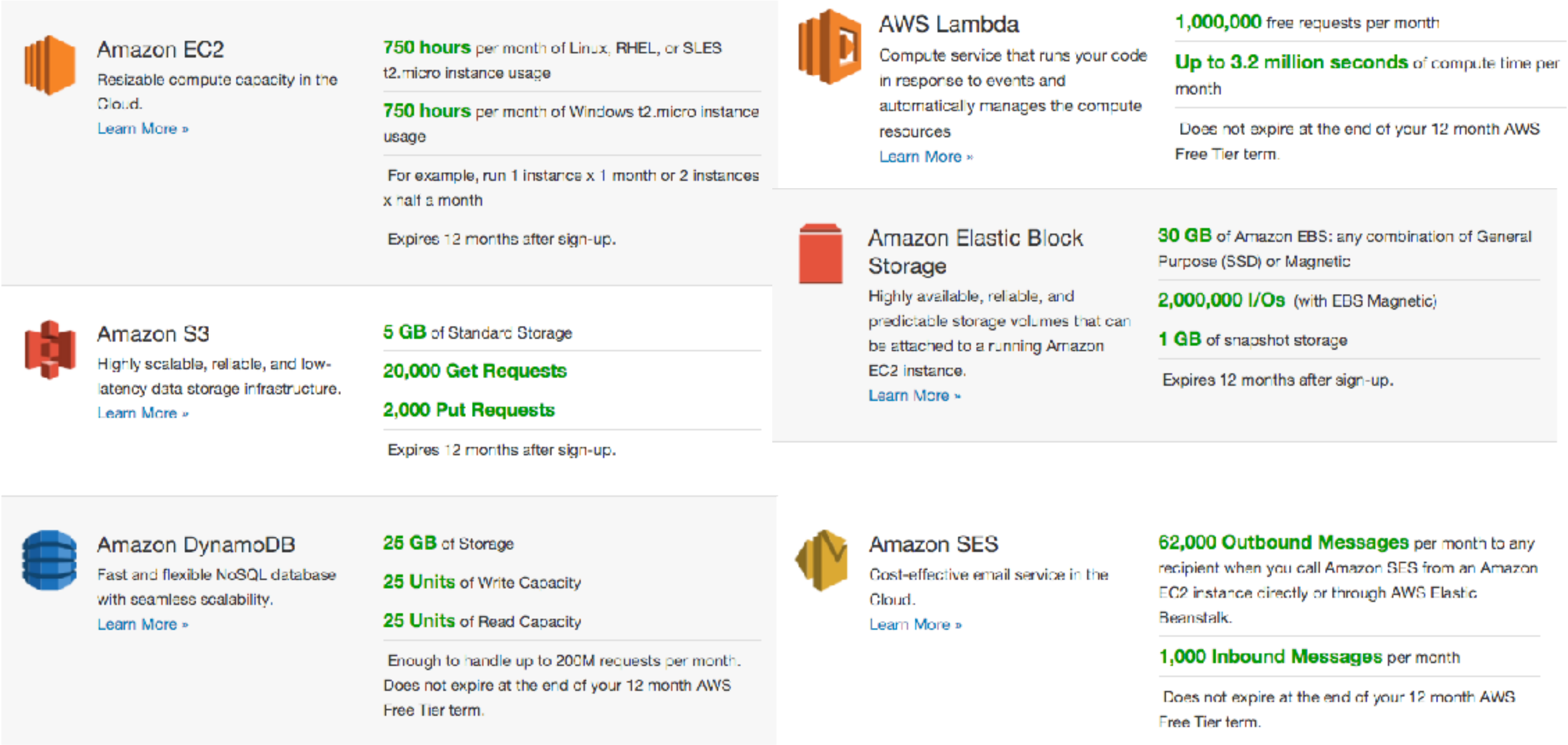
✓ All services operating normally.

Updated: Nov 29 2015 14:20:00 GMT-0500

Service Health Dashboard

AWS Free Tier

- New AWS accounts are entitled to $750/month of "free tier" service.
- Includes micro instances (typically bill at $0.015 cents/hour)
- **Does not include EMR or large machines.**



Amazon EC2

Resizable compute capacity in the Cloud.
Learn More »

750 hours per month of Linux, RHEL, or SLES t2.micro instance usage

750 hours per month of Windows t2.micro instance usage

For example, run 1 instance x 1 month or 2 instances x half a month

Expires 12 months after sign-up.

Amazon S3

Highly scalable, reliable, and low-latency data storage infrastructure.
Learn More »

5 GB of Standard Storage

20,000 Get Requests

2,000 Put Requests

Expires 12 months after sign-up.

Amazon DynamoDB

Fast and flexible NoSQL database with seamless scalability.
Learn More »

25 GB of Storage

25 Units of Write Capacity

25 Units of Read Capacity

Enough to handle up to 200M requests per month. Does not expire at the end of your 12 month AWS Free Tier term.

AWS Lambda

Compute service that runs your code in response to events and automatically manages the compute resources
Learn More »

1,000,000 free requests per month

Up to 3.2 million seconds of compute time per month

Does not expire at the end of your 12 month AWS Free Tier term.

Amazon Elastic Block Storage

Highly available, reliable, and predictable storage volumes that can be attached to a running Amazon EC2 instance.
Learn More »

30 GB of Amazon EBS: any combination of General Purpose (SSD) or Magnetic

2,000,000 I/Os (with EBS Magnetic)

1 GB of snapshot storage

Expires 12 months after sign-up.

Amazon SES

Cost-effective email service in the Cloud.
Learn More »

62,000 Outbound Messages per month to any recipient when you call Amazon SES from an Amazon EC2 instance directly or through AWS Elastic Beanstalk.

1,000 Inbound Messages per month

Does not expire at the end of your 12 month AWS Free Tier term.

—*Some of these expire after 12 months, some don't.*

- https://aws.amazon.com/free/

Amazon EC2

EC2 — Elastic Compute Cloud

Virtual Machines in the cloud — You create "Instances"

- Horizontal Scaling — Create *many VMs.*
- Vertical Scaling — Create small and large VMs (cores, RAM, networking)
- Geographical Diversity — Create in different physical locations ("availability zones")

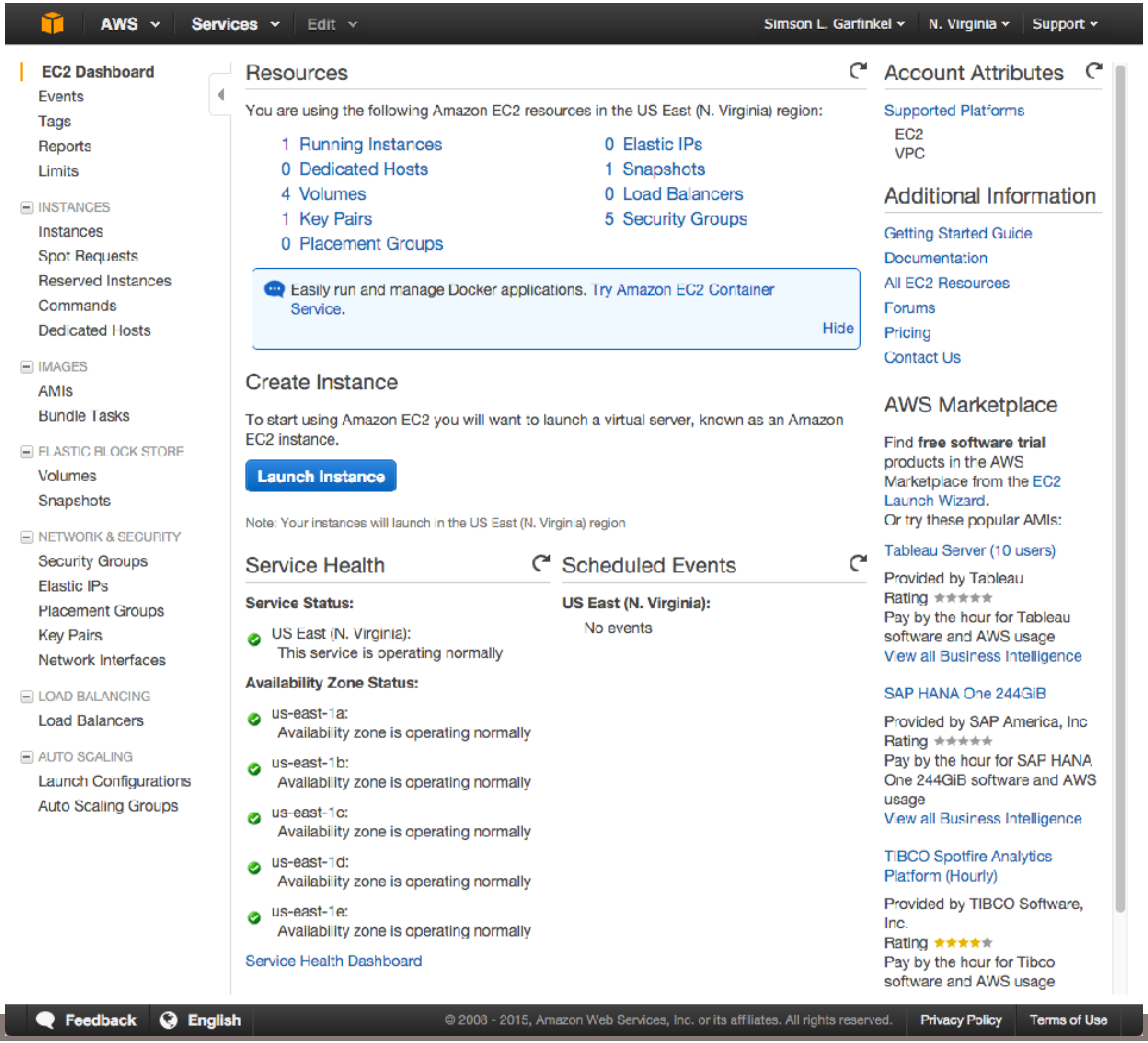
Each instance has:

- **AMI** — Amazon Machine Image — the initial "boot volume"
- Network interface and firewall

Storage:

- **Virtual drives** — Elastic Block Store; can survive shut-down.
- **Physical drives** — Part of the instance; dies with when VM terminates.

EC2 Dashboard



Resources

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

1 Running Instances	0 Elastic IPs
0 Dedicated Hosts	1 Snapshots
4 Volumes	0 Load Balancers
1 Key Pairs	5 Security Groups
0 Placement Groups	

Easily run and manage Docker applications. Try Amazon EC2 Container Service.

Hide

Create Instance

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

Launch Instance

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

Service Health

Service Status:

US East (N. Virginia):
This service is operating normally

Availability Zone Status:

us-east-1a:
Availability zone is operating normally

us-east-1b:
Availability zone is operating normally

us-east-1c:
Availability zone is operating normally

us-east-1d:
Availability zone is operating normally

us-east-1e:
Availability zone is operating normally

Service Health Dashboard

Scheduled Events

US East (N. Virginia):

No events

Account Attributes

Supported Platforms
EC2
VPC

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

AWS Marketplace

Find **free software trial** products in the AWS Marketplace from the EC2 Launch Wizard.
Or try these popular AMIs:

Tableau Server (10 users)
Provided by Tableau
Rating ★★★★★
Pay by the hour for Tableau software and AWS usage
View all Business Intelligence

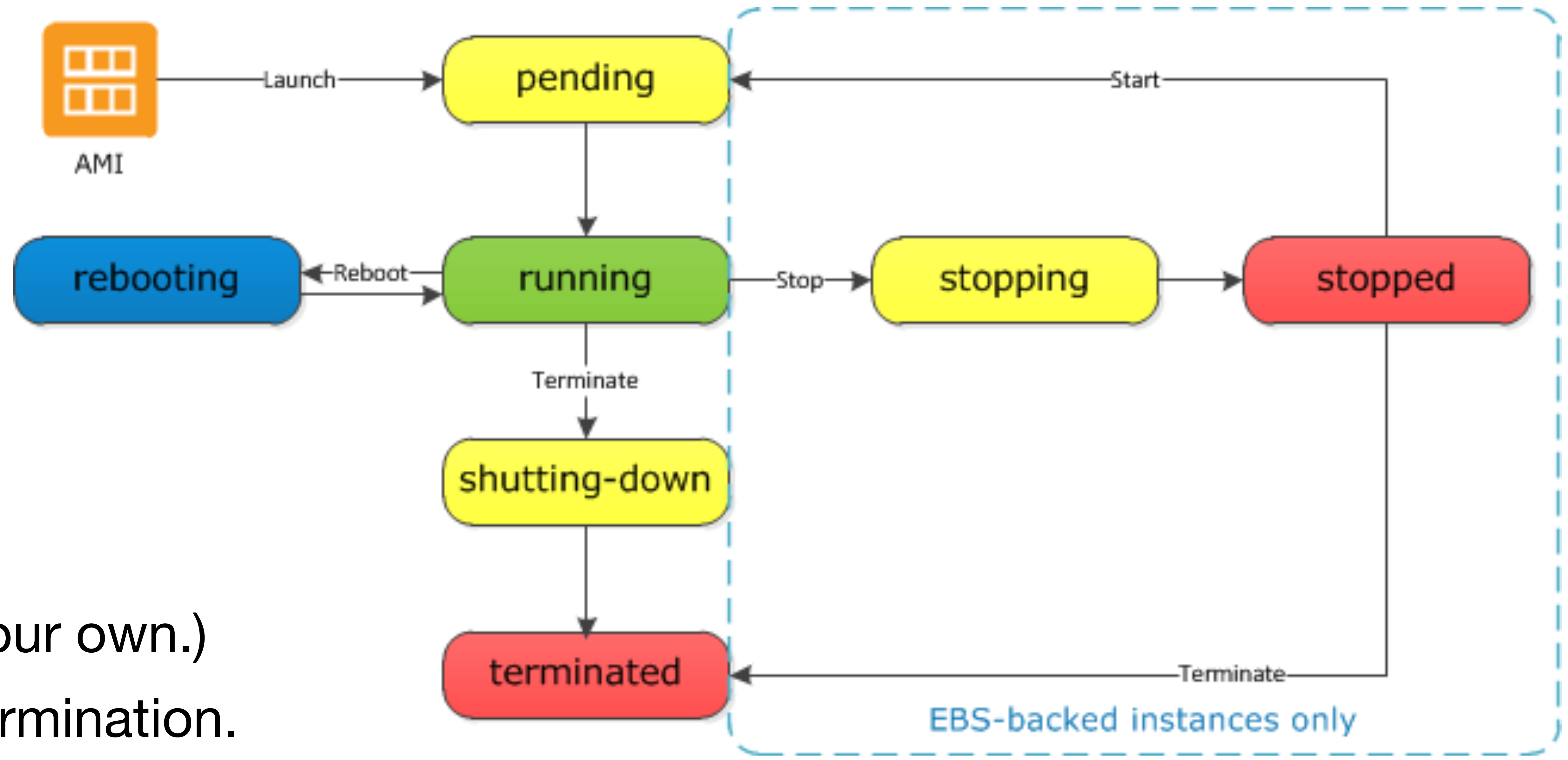
SAP HANA One 244GiB
Provided by SAP America, Inc
Rating ★★★★★
Pay by the hour for SAP HANA One 244GiB software and AWS usage
View all Business Intelligence

TIBCO Spotfire Analytics Platform (Hourly)
Provided by TIBCO Software, Inc.
Rating ★★★★★
Pay by the hour for Tibco software and AWS usage

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Instance life cycle:

- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html

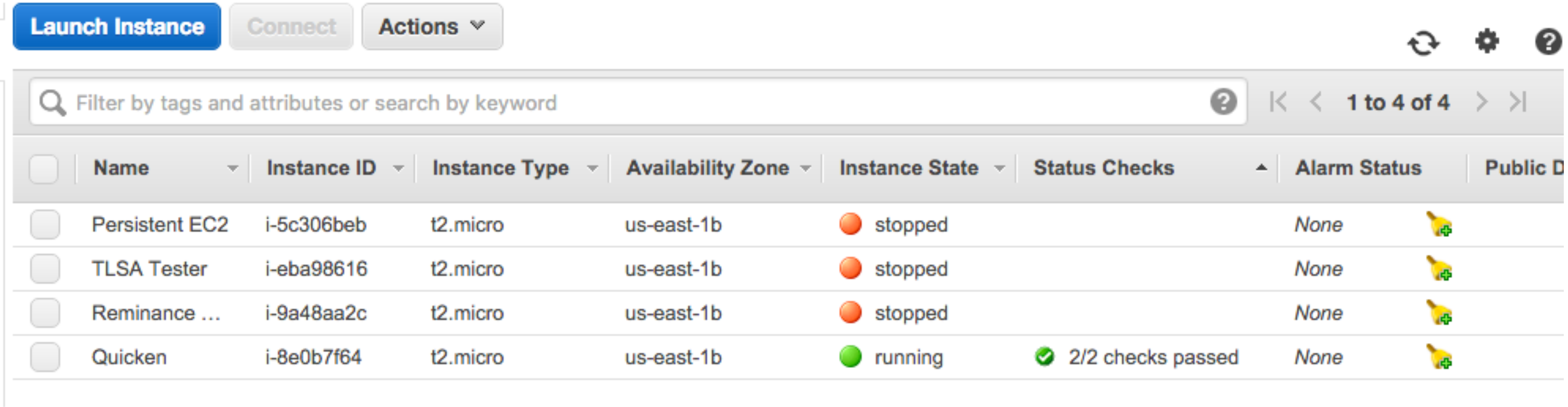


- All instances boot from an AMI (you can upload your own.)
- You specify if the EBS volume is kept or lost on termination.

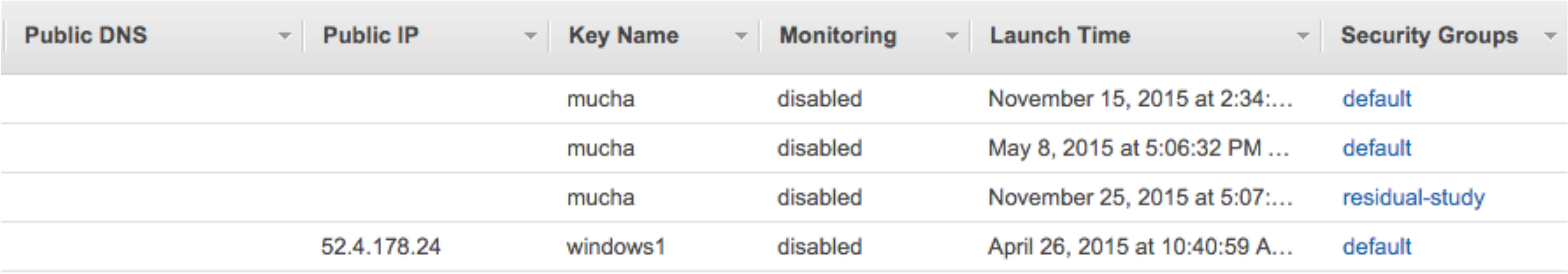
You pay for:

- Instances that are running
- EBS-backed storage
- Bandwidth from EC2➜Rest of Internet

Instance control panel:



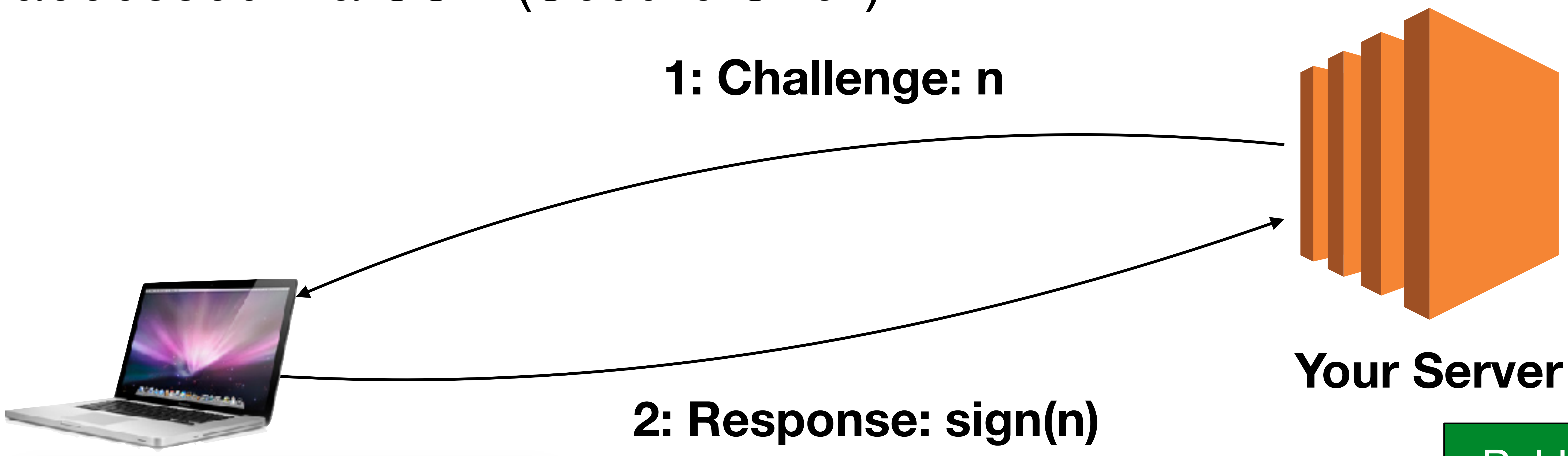
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public D
□	Persistent EC2	i-5c306beb	t2.micro	us-east-1b	● stopped		*None*	
□	TLSA Tester	i-eba98616	t2.micro	us-east-1b	● stopped		*None*	
□	Reminance ...	i-9a48aa2c	t2.micro	us-east-1b	● stopped		*None*	
□	Quicken	i-8e0b7f64	t2.micro	us-east-1b	● running	✓ 2/2 checks passed	*None*	



Public DNS	Public IP	Key Name	Monitoring	Launch Time	Security Groups
		mucha	disabled	November 15, 2015 at 2:34:...	default
		mucha	disabled	May 8, 2015 at 5:06:32 PM ...	default
		mucha	disabled	November 25, 2015 at 5:07:...	residual-study
	52.4.178.24	windows1	disabled	April 26, 2015 at 10:40:59 A...	default

Accessing your instance:
AWS key pairs

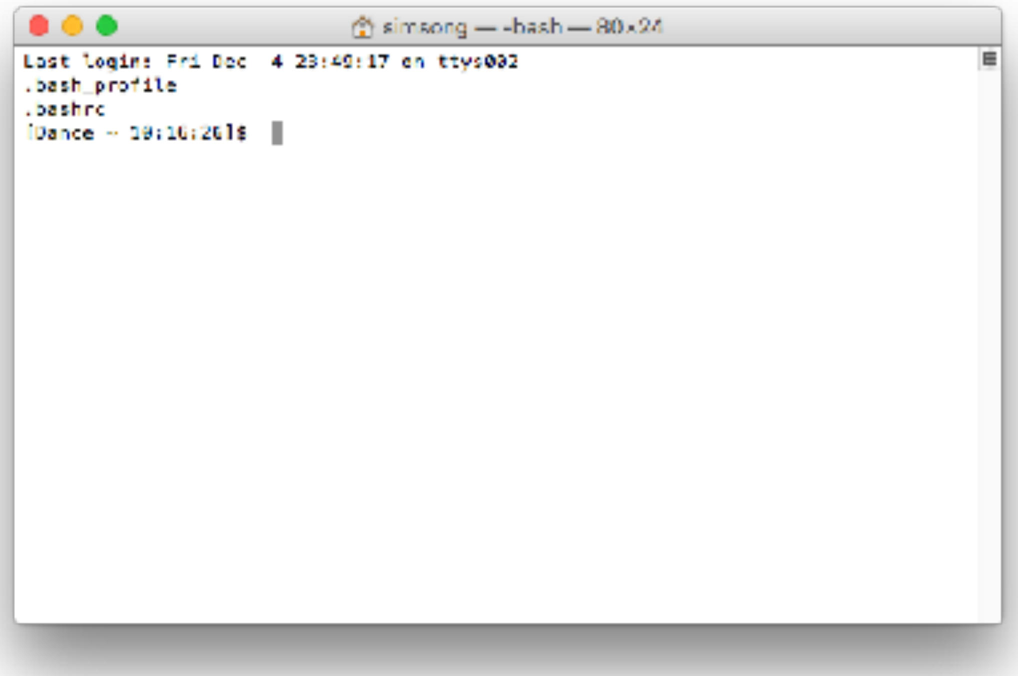
Linux instances are accessed via SSH (Secure Shell)



1: Challenge: n

2: Response: sign(n)

Your Server



```
Last login: Fri Dec  4 23:40:17 on ttys002
.bash_profile
.bashrc
[Dance ~ 10:16:26]$
```

Public key 1

Secret key 1

Public key 1

3: Verify(sign(n))

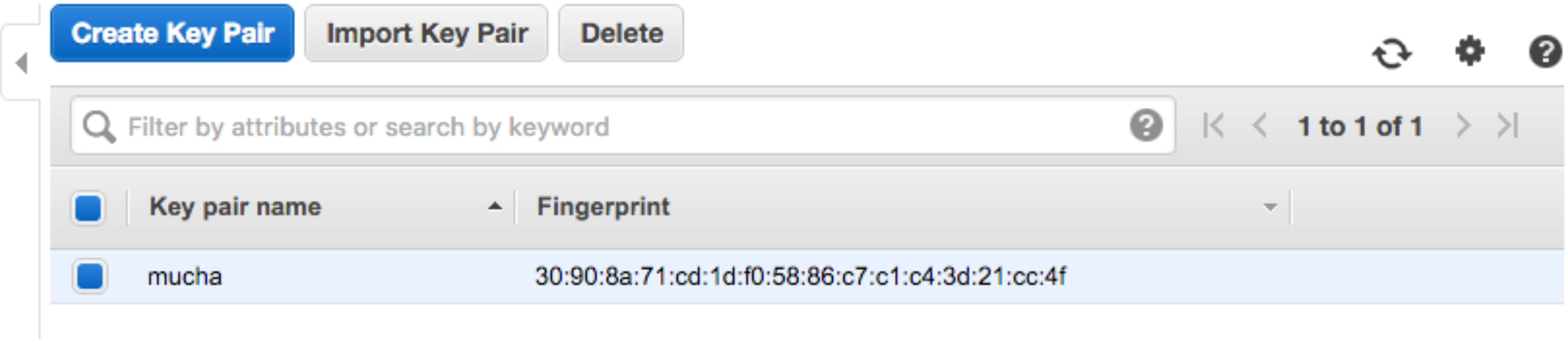
Accessing your instance:
AWS key pairs

Linux instances are accessed via SSH (Secure Shell)

- AWS uses SSH "public key authentication."
- Two ways to get your public key to Amazon:
  - —*You create a public/private keypair with "ssh keygen -t rsa -f mykey.pem" & import*
  - —*Amazon will create the pair and you download it.*
- You use the private key to authenticate.

Key pairs:

- Each key is identified by a "Fingerprint."
- If you lose your private key, you can't access your server.



■	Key pair name	Fingerprint
■	mucha	30:90:8a:71:cd:1d:f0:58:86:c7:c1:c4:3d:21:cc:4f

Making a key…

```
$ ssh-keygen -t rsa -f mykey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in mykey.
Your public key has been saved in mykey.pub.
The key fingerprint is:
SHA256:B+/MiY/KgrDy5Agc8pkP+/AKd7YbA2Gdno7PnnQmfXs simsong@Dance.local
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|   . .           |
|  o o   .        |
| . o .   o       |
|... o   S o      |
|+..* .   * .     |
|o+X.O + o =      |
|+*.%.X . +E      |
|o.=+@oo.o..      |
+----[SHA256]-----+
$ cat mykey.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDKvpBsCUJBGxJiGtvQF6f3LrRWpk5EbsWCDkZ2lMAMdVg0Ee4iRUuKfo62KXge8GgOuOzSSG43L/yvn+
+LtV4s8sNYA1QtxDyZmFCYLGV0slRxkKL/xN/
KwxNc5EgiP1tVNrQvAhrKUCIQDspNuDPbO5DvGxb+YyJdUAW5X5Z3DmGaylJotM0ypMaqE5+xHQndiusg9YIy7B8xFhoKCJ5+B+HlQdiQUQULuTlD2oSxLd0Wd5MIF/
OaZ+0uu9HujqDWc5TweNcHPt3ycS//s9ITNhjoUddCd3gHCH3TH5rZwM79MpAtCZipyKvowvFjgDqvAdt6MlvULQ7wpJKT9+Tl simsong@Dance.local
$
```

The matching private key…

```
$ cat mykey
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,F5A1FC20F3E4F8B7BECFE707F8972509

7Ac+l4q3dfJ+0vBLdgWx3LdxV0vAF9YuWPIZqJjm1CzjLMM1POMk9Juj4PdwvlSh
btZZ+h1zymbgcdrTd2ivALIkO0PbjlgZpnZ9Vkn4MXEjsOPq/Xa2B+tI4/i3HpUL
cCBcqwNujt6pgzoVCiv5L+33Hiiupn9d9NZbdDxWTyArzYAU7vwfb6UejGrZ8ME1
mYl3qYsBtOGhViY8AkDidTYY40t4Sz3Yk66a3ZiTcBivnOBpdIgZ1S8t3Stoaprt
jrlm5eZFDtA7hIlXdNBsgu8xmKDyGAx/bP7WFakCsoCsUQOyl53oPZcIFkivyVqV
dki8MCiGZAu0oh68L2qzzINjNfOGQCBQIosxLMWCAT5KRyGMzuxmARxgc6ZWmabJ
xwSgHT+UdNq3IEP8k/5GPxHsbmVdWe6MXmVfSziZFAPYqEYI/FqiiNVoZDMOR2pf
sSImx2DtYUhe5c4b5n3I20wEnY9gDOkzDjiW4nJ/l4Tk2HXX0yPIMeFlLbHdisU2
FPWp85x43Ibeesd/oTOwRUiPE/4a8y0QNcdsHOjefiWZBL5qgH5vr8tsgYFN5IKv
peJh2DVGQd7u1jnK1hnKIF6+TbvnQmx9RPrVX3Nab1ba9s3B539AGxIhmyoKXyEG
UwbgPYXHRmjCgf8ENulbREdvhBV99cXldk6RlsCoHZjzE1FGKiPOGpHd2fqj2PTp
ZFOXQZ2Xdnl/L26wzr5M9daltlUfa7rSllrpQFDQ2OP0LOGvLVVrmHRYmwUgv2Fb
BbxgwVtOp6MNAxNz0uqjkG+LO44GBV93eVv8aT/2s4V0/s4ul13uGDFqT+NOwYz3
/SA83Mg6u67Aoqpb0261ieeUeaalx+NECfY904tlLtnnYsKnbx6FiThgQXTxRf7R
iXpUQNCkNG0CAaKb4jdlNj2vb99VRfW42Ldoh6pGWCCXAXRmJ5O18s+bqySfSYGP
+n+f45+yPZRbnxGujZRWOZ/apmCcVyNIBsv3+3smW6ISz7jtXZPFxxRcrOnUr2VT
YnUgDJWskwB+aJpPn8KbvOijjlTOi5k3KgdOjVgTvzLhuOsCPrxPnuET8LG/e5PZ
sq3vsn2hWhcqKDSmzyXL0iFkAqByOxIh5hLMQJ5yGz3RnH9yJEBv8xpXFISjxgmg
op2HwN2e08HuZZQ2gRAIdZgTOJVD+hTv+fbBpV+sTGVLqcVxxjHAOX1WvZgeB2Ax
EH0fXBjthyas9GBjJ3EQtHpcFKQpj+HPX0IdkpYn35BSED3I9mn1OeWcLugvMb9l
OUHTcyw5Gi2sAxdrxNTt9XXFsNiSdxkdPlQHqJe94KilFrPWgYBVj8c39fynI8qL
/N0NOZS/S/FCdua05wIF40LuZzTqtkB5A7CYZinRQiBiTGNJAq0uB7wkU6gu+woj
t33rN6cuoim/SNxQiyJhSgHLF4nRMY+z6Yly7x6sZCBgUJcqvxFyJlhFc13fL4JE
tXAIhyiDV0e8fkr7+yGw6firlUuV1X+eZG4SDAD109phhsdRKjIEw+QBfpE8o7B1
vkRdkjAofY5rm3kzjxnInjbT1FXnwo9r6iIbJ3v0ExLRTmjga9UhNdz3qtuc6Bkx
-----END RSA PRIVATE KEY-----
$
```

Private keys can be encrypted or decrypted.

You should always store your keys encrypted.

AWS requires decrypted keys for upload.

Sncrypt & decrypt with ssh-keygen

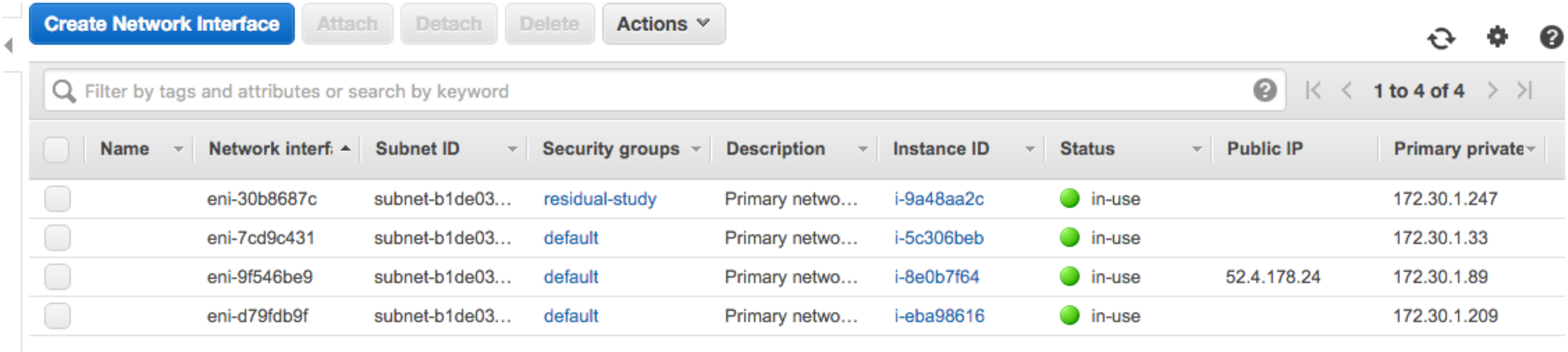
Network interfaces: each instance has 1 “virtual” interface, but possibly 2 IP addresses.

Amazon assigns a private IP address and a public IP address.

- Private IP address is the “real” address on your private subnet.
- Amazon uses two-way NAT to provide the “public” address.
- NAT implements firewall through “security groups.”

Other options:

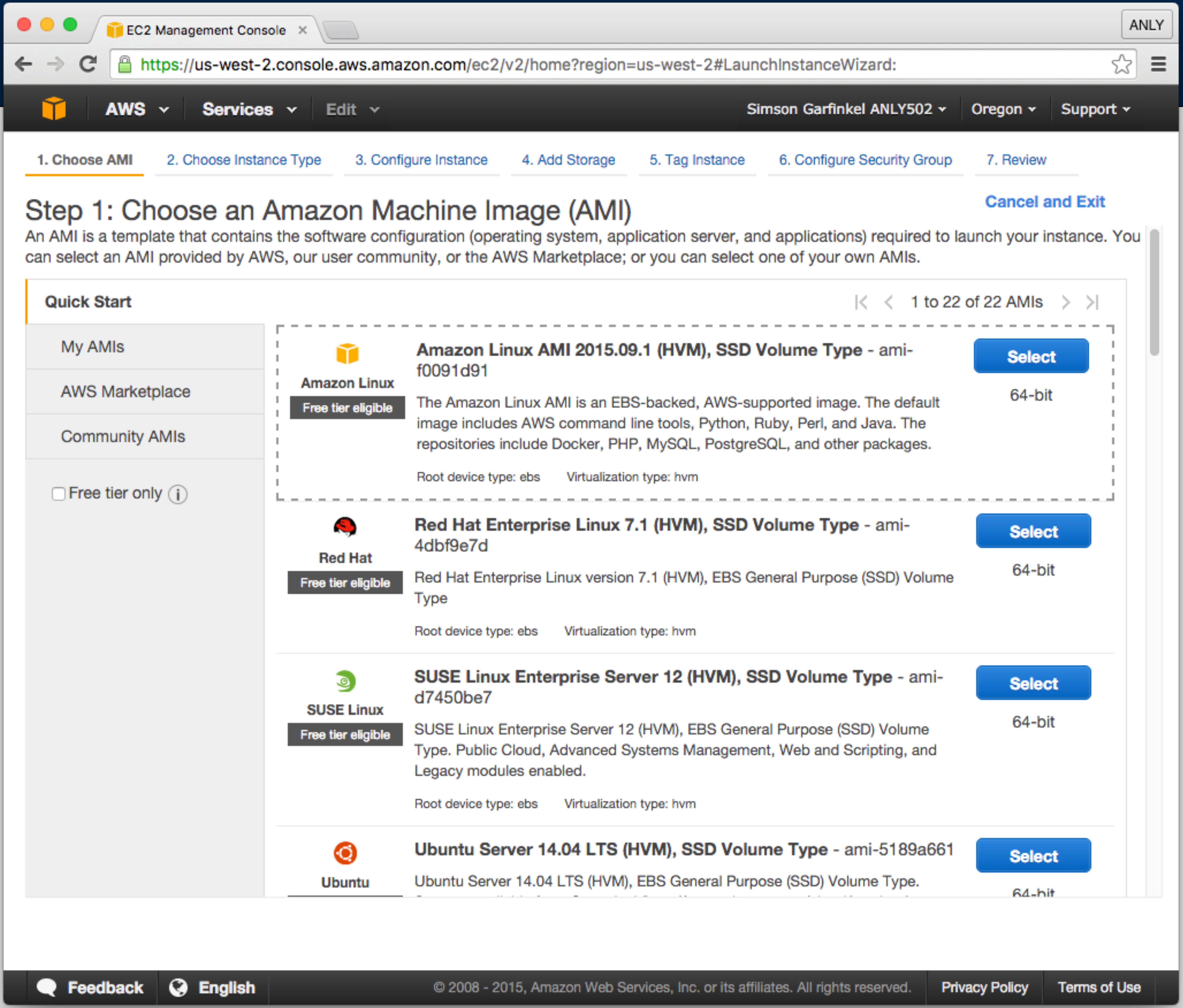
- You can have only private addresses. (More secure.)
- VPN to your organization. (Not in this course.)



□	Name	Network interf.	Subnet ID	Security groups	Description	Instance ID	Status	Public IP	Primary private
□		eni-30b8687c	subnet-b1de03...	residual-study	Primary netwo...	i-9a48aa2c	● in-use		172.30.1.247
□		eni-7cd9c431	subnet-b1de03...	default	Primary netwo...	i-5c306beb	● in-use		172.30.1.33
□		eni-9f546be9	subnet-b1de03...	default	Primary netwo...	i-8e0b7f64	● in-use	52.4.178.24	172.30.1.89
□		eni-d79fdb9f	subnet-b1de03...	default	Primary netwo...	i-eba98616	● in-use		172.30.1.209

Putting it all together…

We recommend Amazon Linux



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.

Amazon Linux — Free tier eligible

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d91

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

Red Hat — Free tier eligible

Red Hat Enterprise Linux 7.1 (HVM), SSD Volume Type - ami-4dbf9e7d

Red Hat Enterprise Linux version 7.1 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm

Select — 64-bit

SUSE Linux — Free tier eligible

SUSE Linux Enterprise Server 12 (HVM), SSD Volume Type - ami-d7450be7

SUSE Linux Enterprise Server 12 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs Virtualization type: hvm

Select — 64-bit

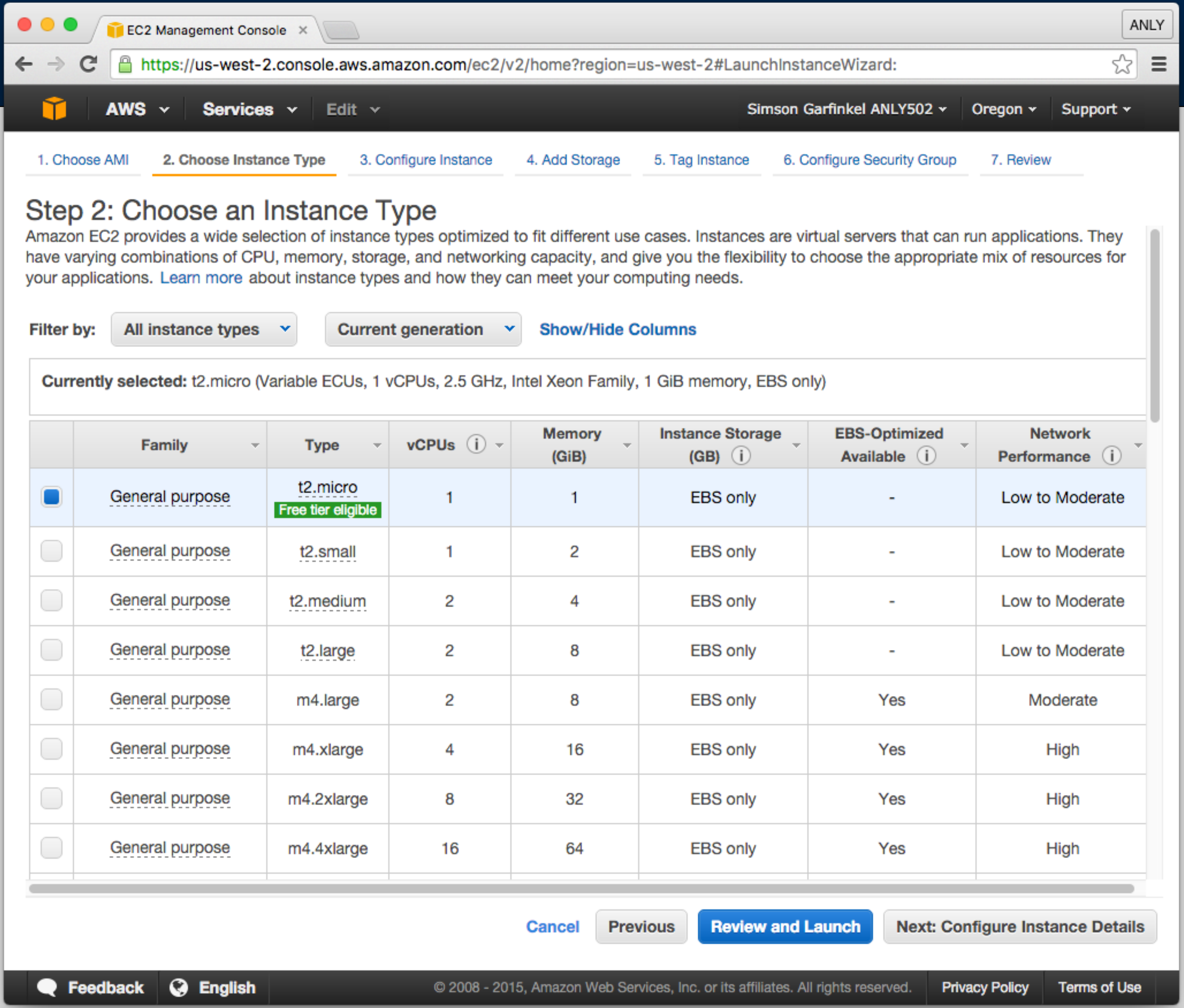
Ubuntu

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-5189a661

Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type.

Select — 64-bit

Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support

1. Choose AMI · 2. Choose Instance Type · 3. Configure Instance · 4. Add Storage · 5. Tag Instance · 6. Configure Security Group · 7. Review

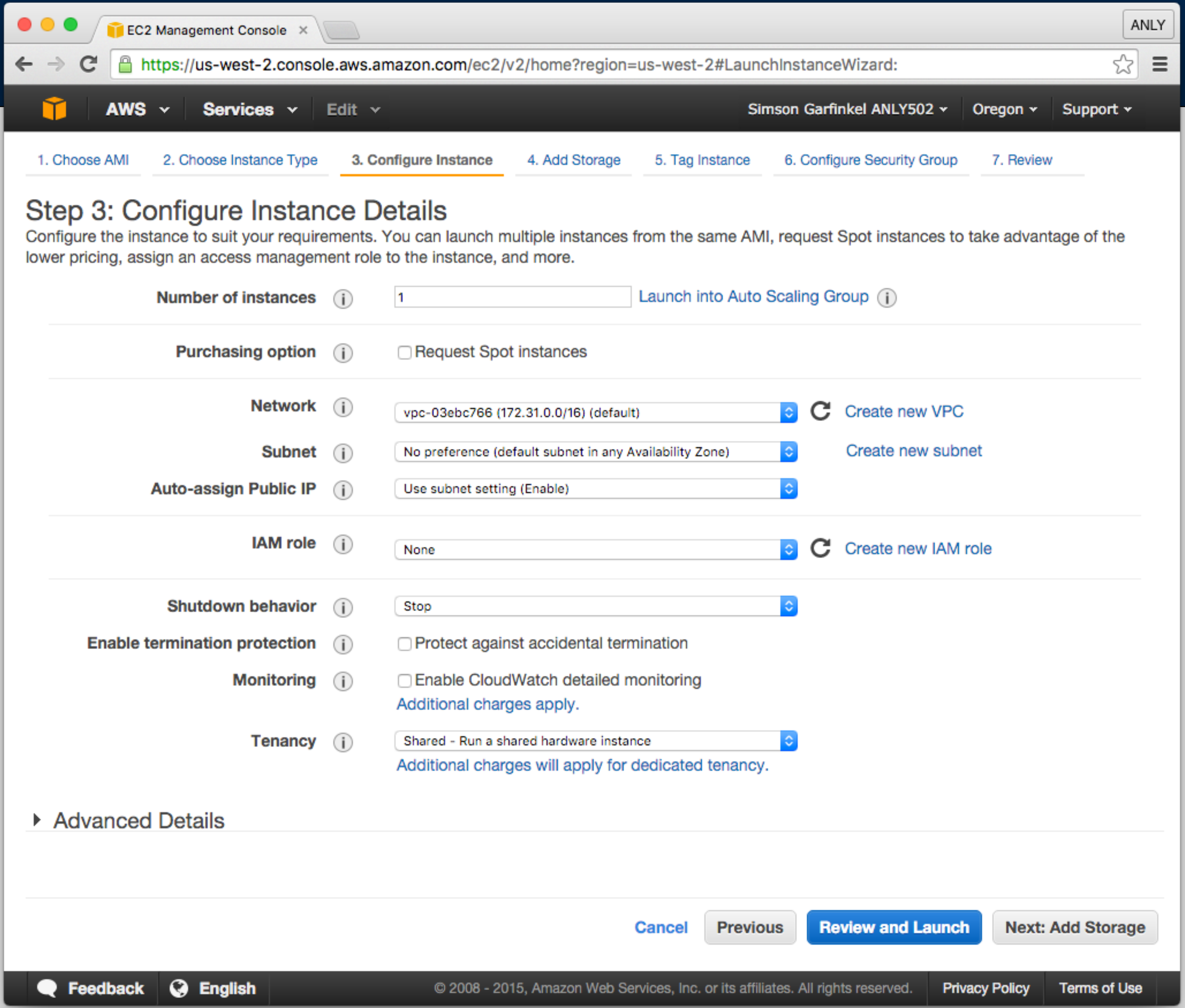
Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types · Current generation · Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate
□	General purpose	m4.xlarge	4	16	EBS only	Yes	High
□	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
□	General purpose	m4.4xlarge	16	64	EBS only	Yes	High

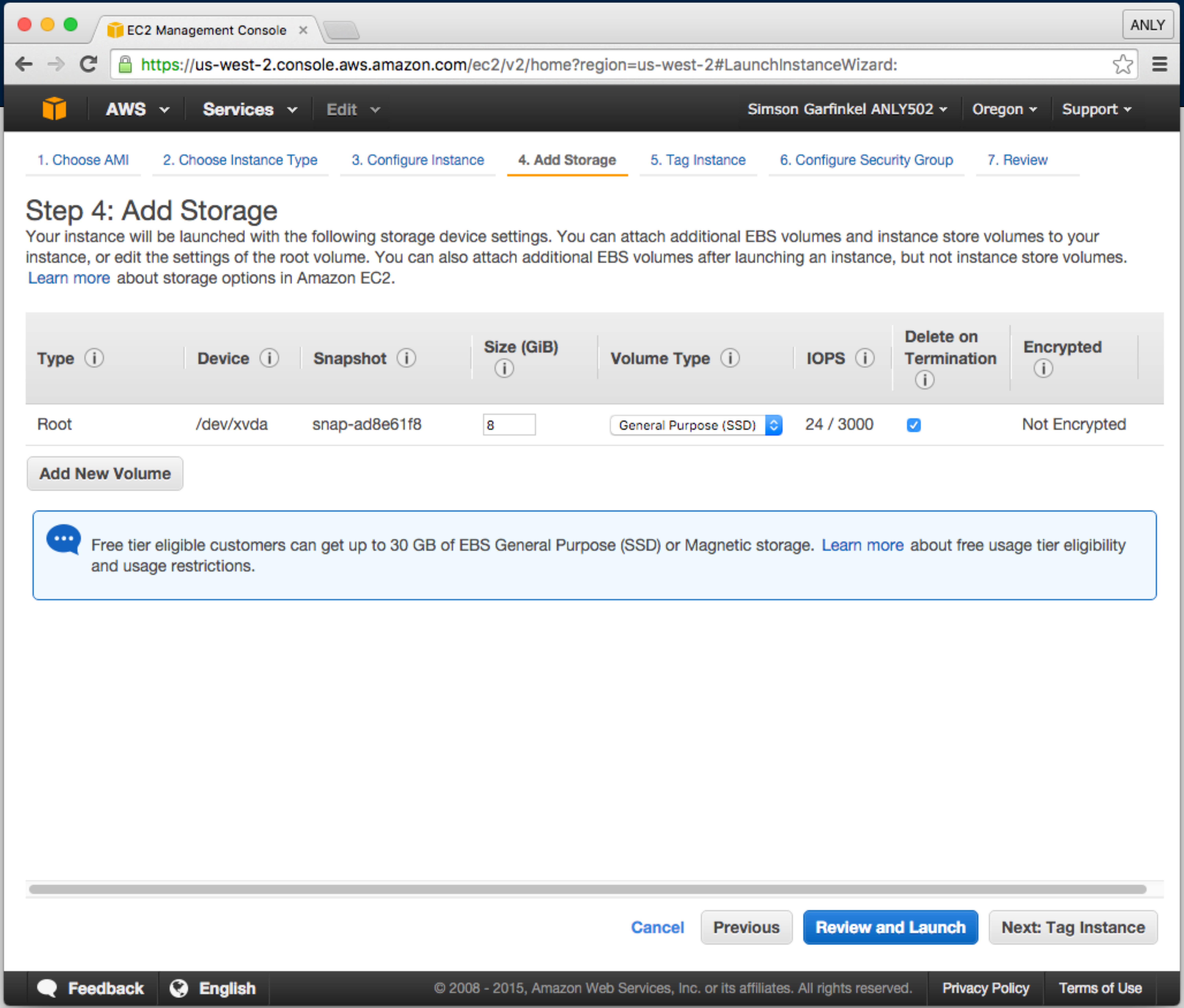


Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances	
Network	vpc-03ebc766 (172.31.0.0/16) (default)	Create new VPC
Subnet	No preference (default subnet in any Availability Zone)	Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
IAM role	None	Create new IAM role
Shutdown behavior	Stop	
Enable termination protection	□ Protect against accidental termination	
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.	
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.	

▸ Advanced Details

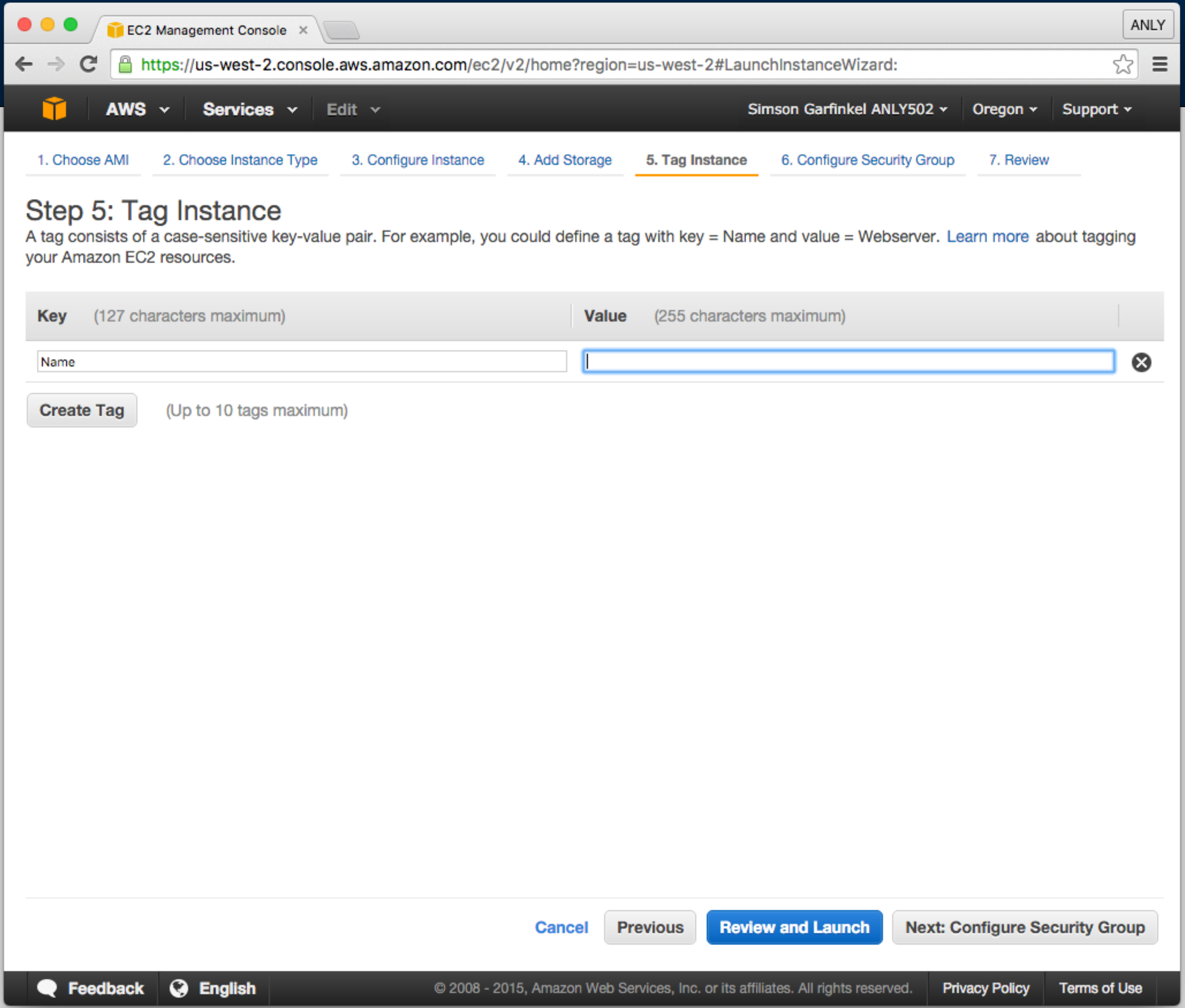


Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-ad8e61f8	8	General Purpose (SSD)	24 / 3000	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.



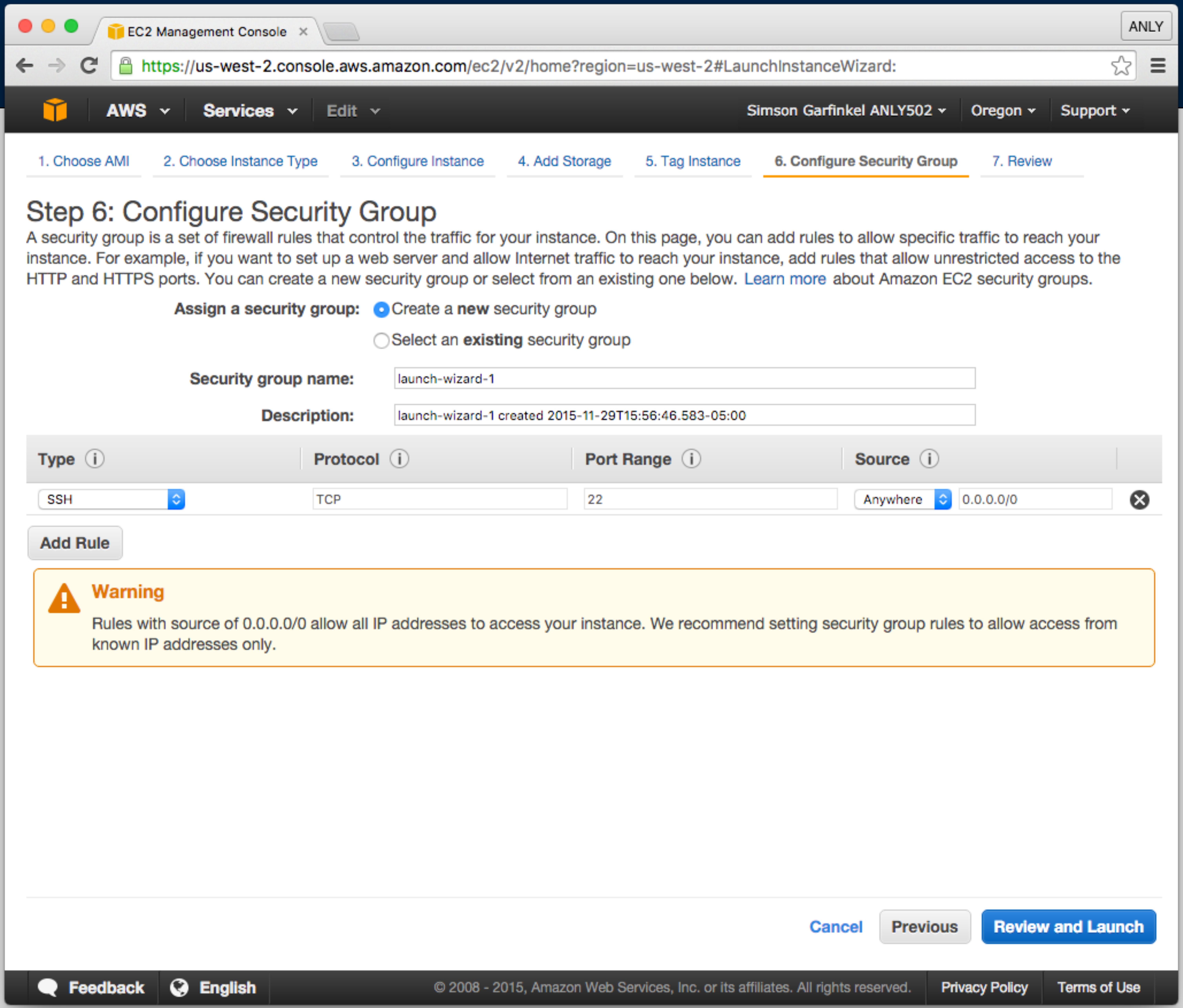
Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	

Create Tag (Up to 10 tags maximum)

Cancel
Previous
Review and Launch
Next: Configure Security Group
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group

○ Select an **existing** security group

Security group name: launch-wizard-1

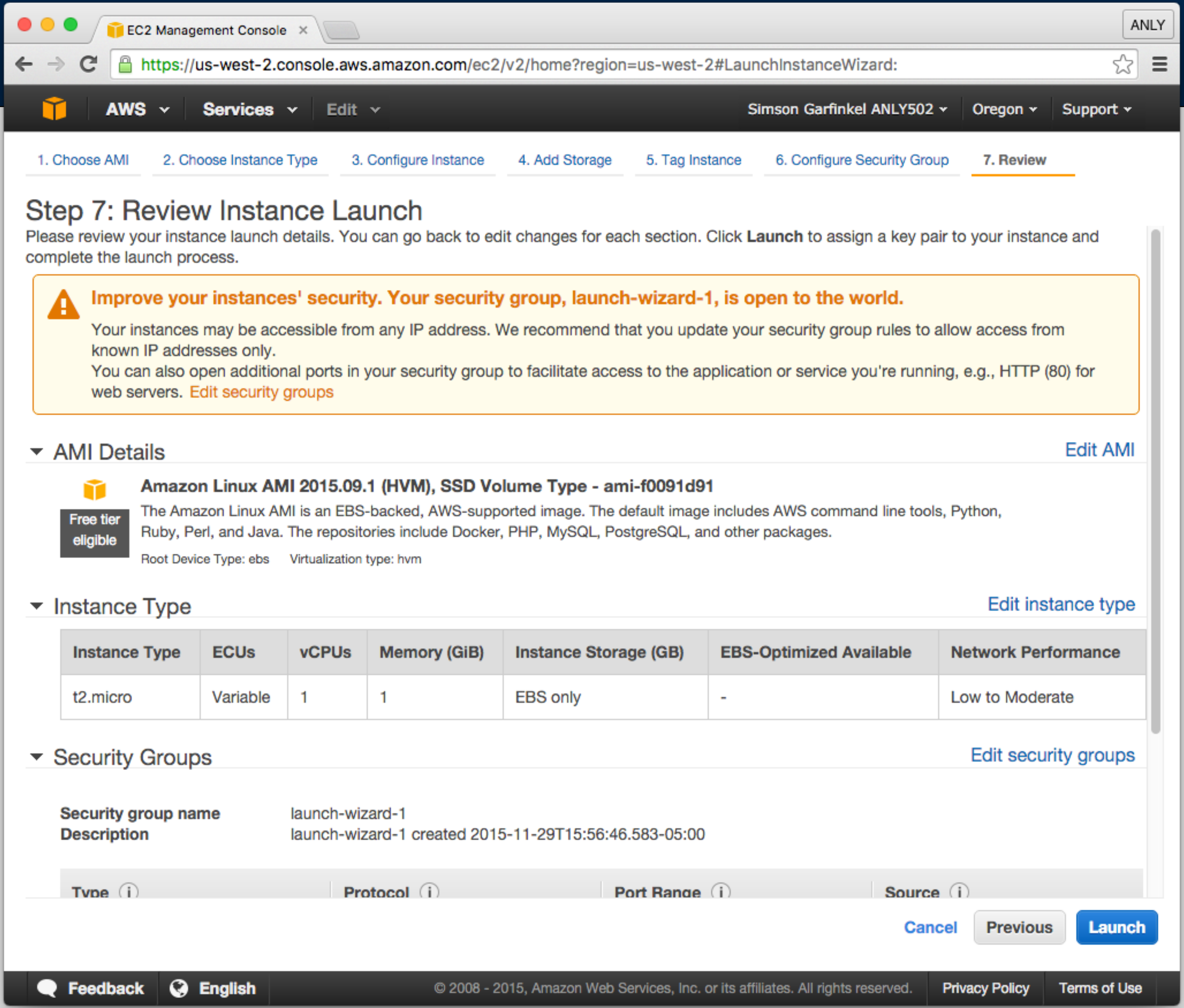
Description: launch-wizard-1 created 2015-11-29T15:56:46.583-05:00

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel
Previous
Review and Launch
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



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

AMI Details

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d91

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

Instance Type

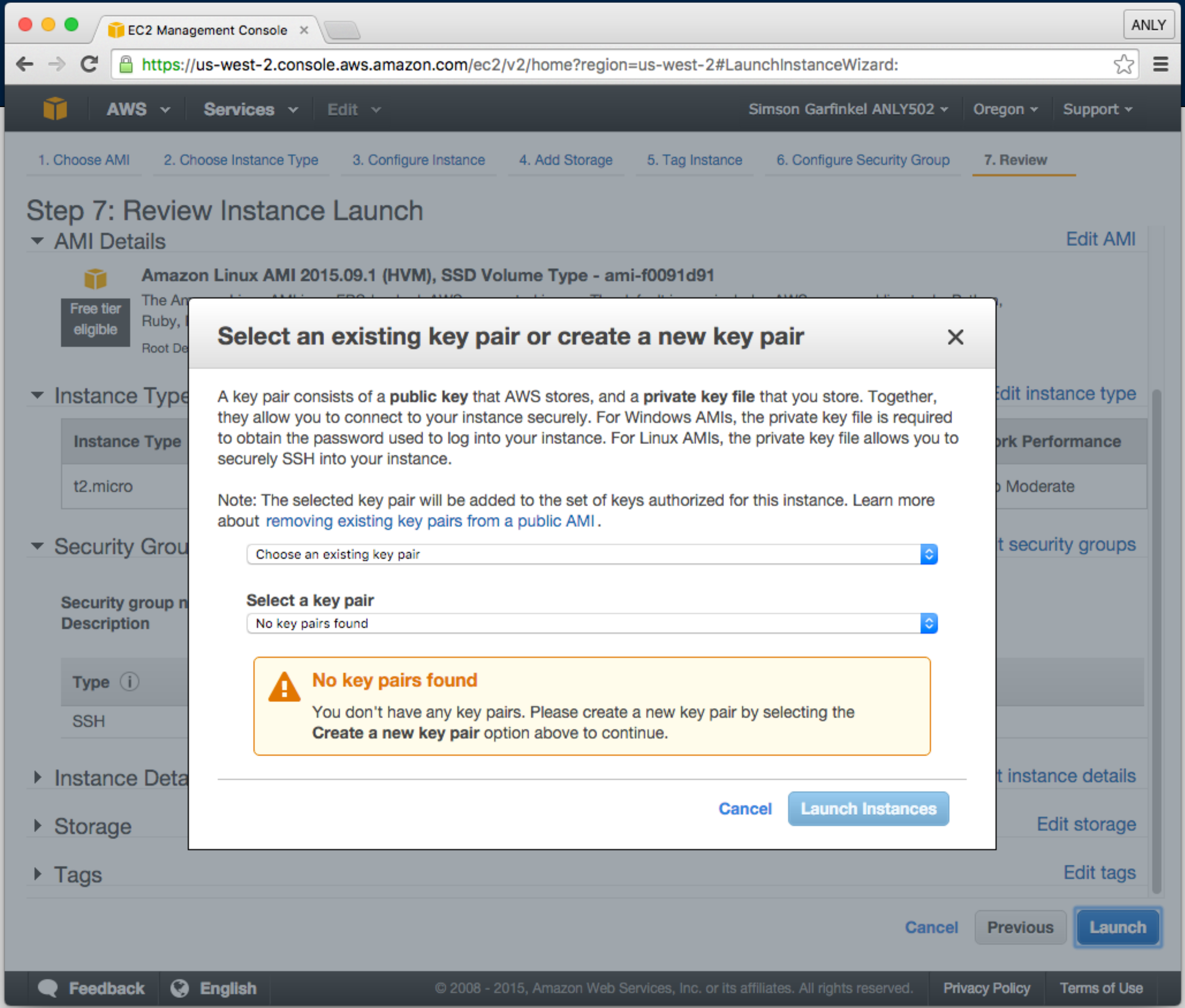
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name launch-wizard-1
Description launch-wizard-1 created 2015-11-29T15:56:46.583-05:00

Type	Protocol	Port Range	Source

Cancel
Previous
Launch
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Step 7: Review Instance Launch

AMI Details

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d91

Instance Type

Instance Type	ork Performance
t2.micro	o Moderate

Security Grou

Security group n
Description

Type
SSH

Instance Deta

Storage

Tags

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

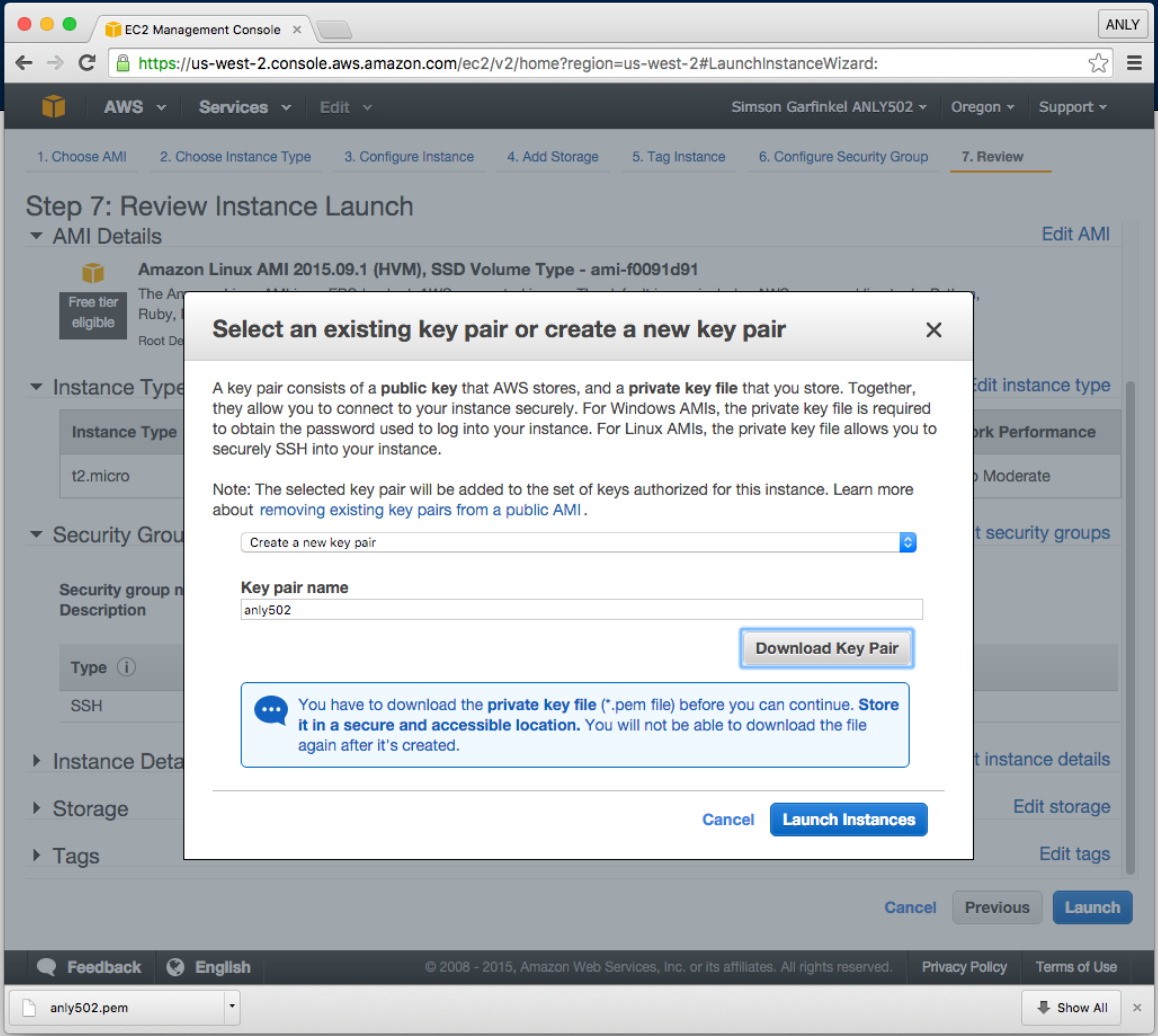
Select a key pair

No key pairs found

No key pairs found

You don't have any key pairs. Please create a new key pair by selecting the **Create a new key pair** option above to continue.

Cancel | Launch Instances



Step 7: Review Instance Launch

AMI Details

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d91

Free tier eligible

Instance Type

t2.micro

Security Group

Type

SSH

Instance Details

Storage

Tags

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

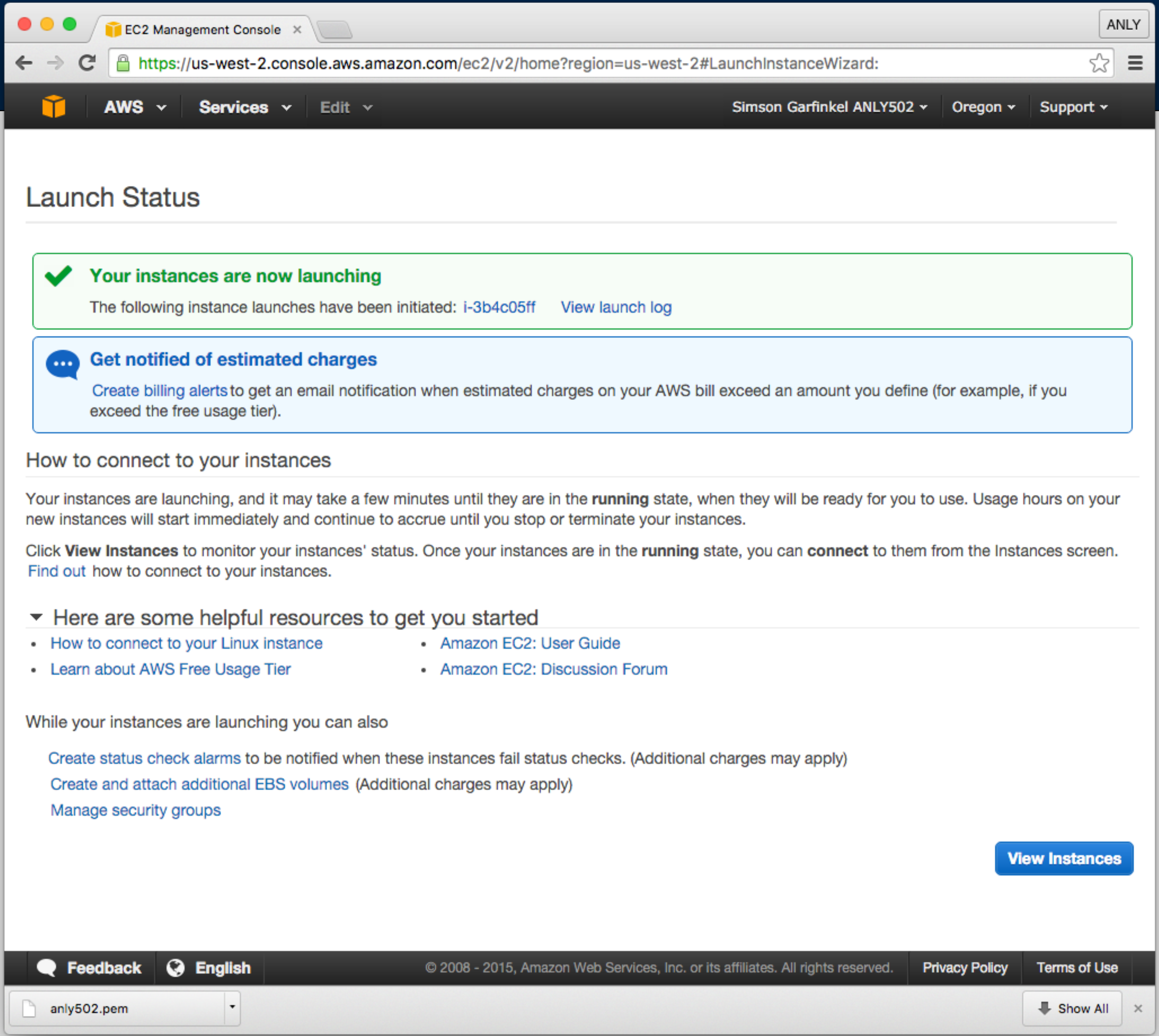
Create a new key pair

Key pair name: anly502

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel | Launch Instances



EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support

Launch Status

✓ **Your instances are now launching**

The following instance launches have been initiated: i-3b4c05ff View launch log

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

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

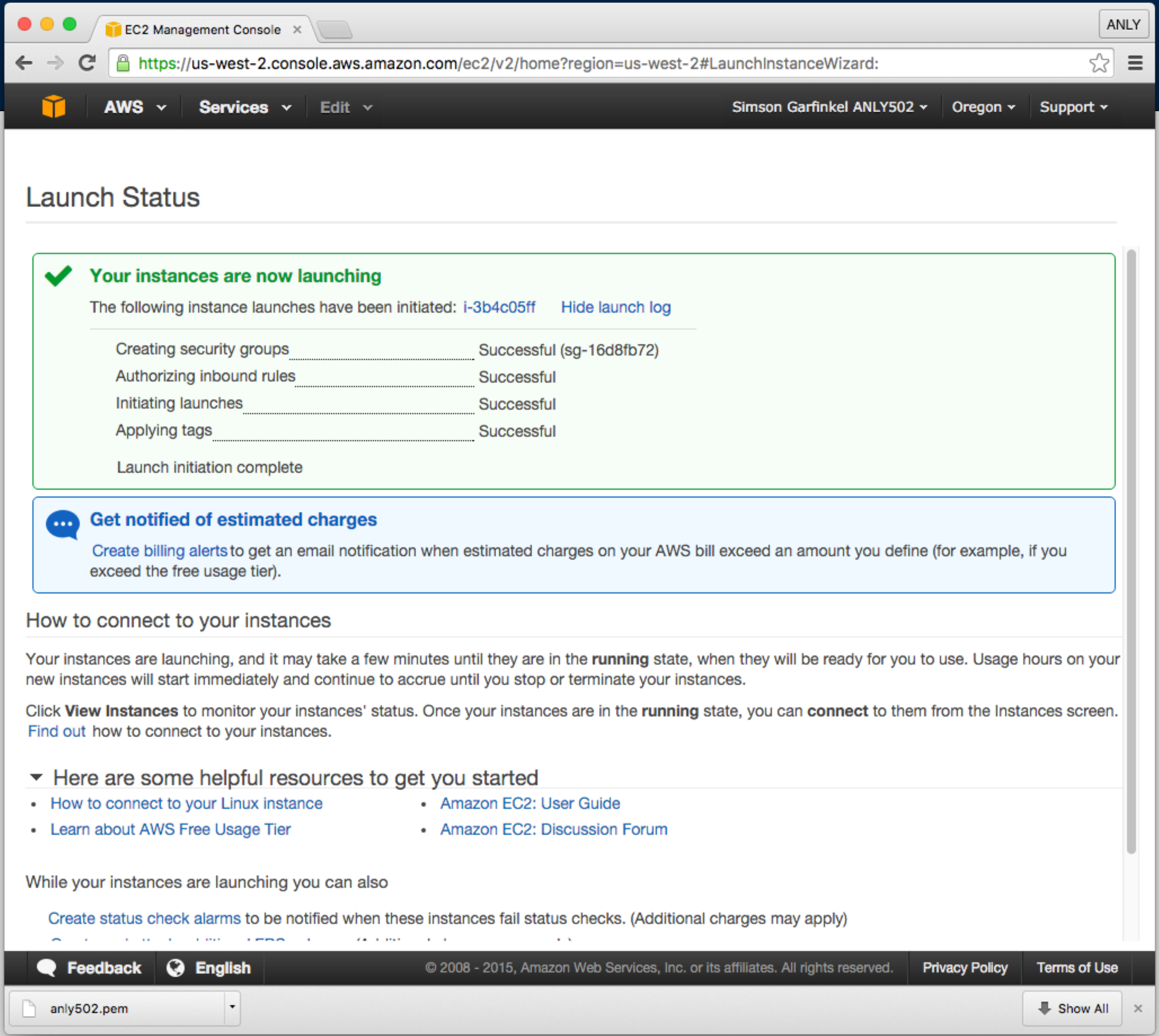
- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

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


EC2 Management Console
ANLY
https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:
AWS
Services
Edit
Simson Garfinkel ANLY502
Oregon
Support

Launch Status

✓ Your instances are now launching

The following instance launches have been initiated: i-3b4c05ff Hide launch log

Creating security groups .... Successful (sg-16d8fb72)
Authorizing inbound rules .... Successful
Initiating launches .... Successful
Applying tags .... Successful

Launch initiation complete

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

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

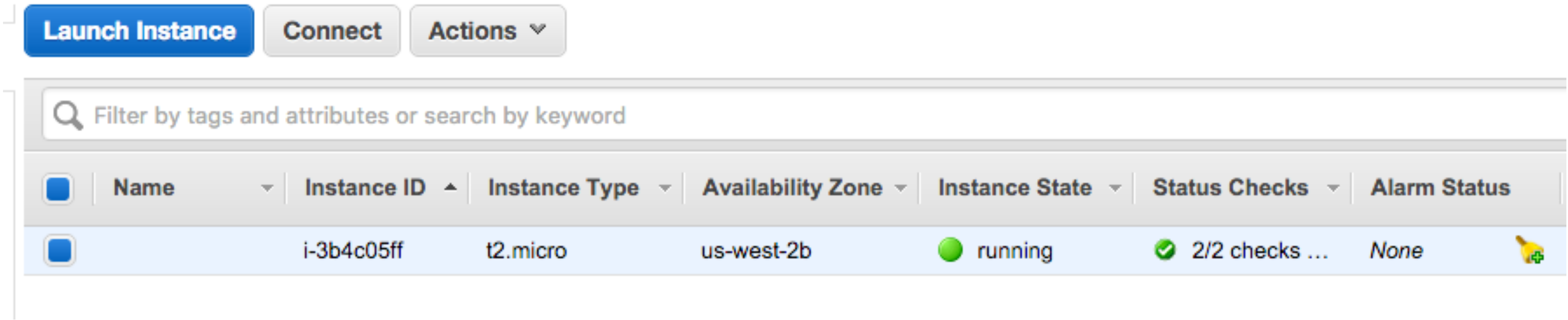
- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

While your instances are launching you can also

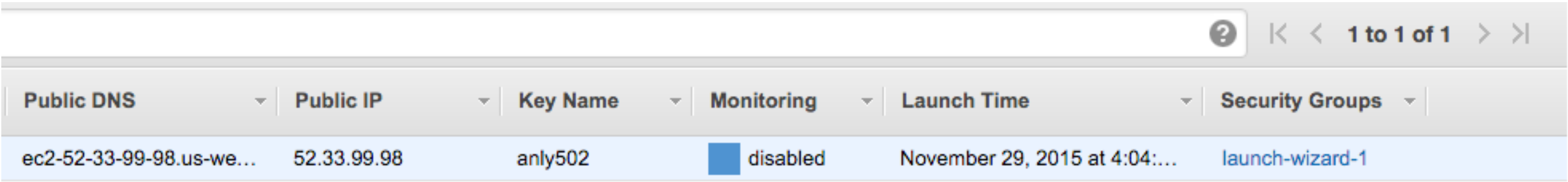
Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)

Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
anly502.pem
Show All

Instance is running…



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
□		i-3b4c05ff	t2.micro	us-west-2b	● running	✓ 2/2 checks …	None



Public DNS	Public IP	Key Name	Monitoring	Launch Time	Security Groups
ec2-52-33-99-98.us-we…	52.33.99.98	anly502	disabled	November 29, 2015 at 4:04:…	launch-wizard-1

Connect…

```
$ ssh -i ~/Downloads/anly502.pem ec2-user@52.33.99.98
The authenticity of host '52.33.99.98 (52.33.99.98)' can't be established.
ECDSA key fingerprint is SHA256:3XZSXZ5AfLYukBFkga243VB9TEoC1mi3VWhNiPlRFcY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.33.99.98' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for '/Users/simsong/Downloads/anly502.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/simsong/Downloads/anly502.pem": bad permissions
Permission denied (publickey).
$
```

Connect…

```
$ ssh -i ~/Downloads/anly502.pem ec2-user@52.33.99.98
The authenticity of host '52.33.99.98 (52.33.99.98)' can't be established.
ECDSA key fingerprint is SHA256:3XZSXZ5AfLYukBFkga243VB9TEoC1mi3VWhNiPlRFcY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.33.99.98' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for '/Users/simsong/Downloads/anly502.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/simsong/Downloads/anly502.pem": bad permissions
Permission denied (publickey).
$
```

Move the private key into place:

```
$ chmod 600 Downloads/anly502.pem
$ mv Downloads/anly502.pem  ~/.ssh/
```

Connect…

```
$ ssh -i ~/Downloads/anly502.pem ec2-user@52.33.99.98
The authenticity of host '52.33.99.98 (52.33.99.98)' can't be established.
ECDSA key fingerprint is SHA256:3XZSXZ5AfLYukBFkga243VB9TEoC1mi3VWhNiPlRFcY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.33.99.98' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for '/Users/simsong/Downloads/anly502.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/simsong/Downloads/anly502.pem": bad permissions
Permission denied (publickey).
$
```

Move the private key into place:

```
$ chmod 600 Downloads/anly502.pem
$ mv Downloads/anly502.pem  ~/.ssh/
```

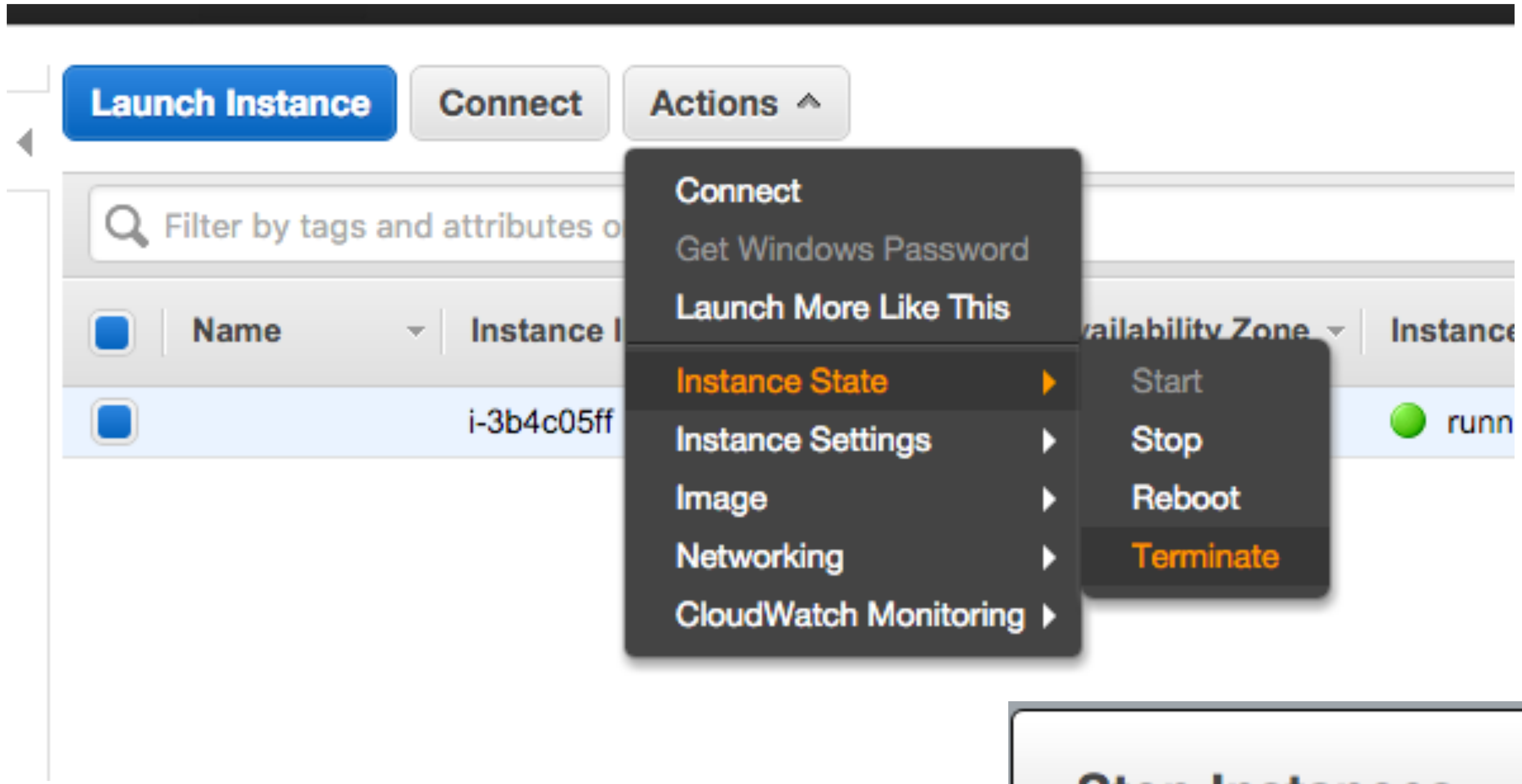
And connect!

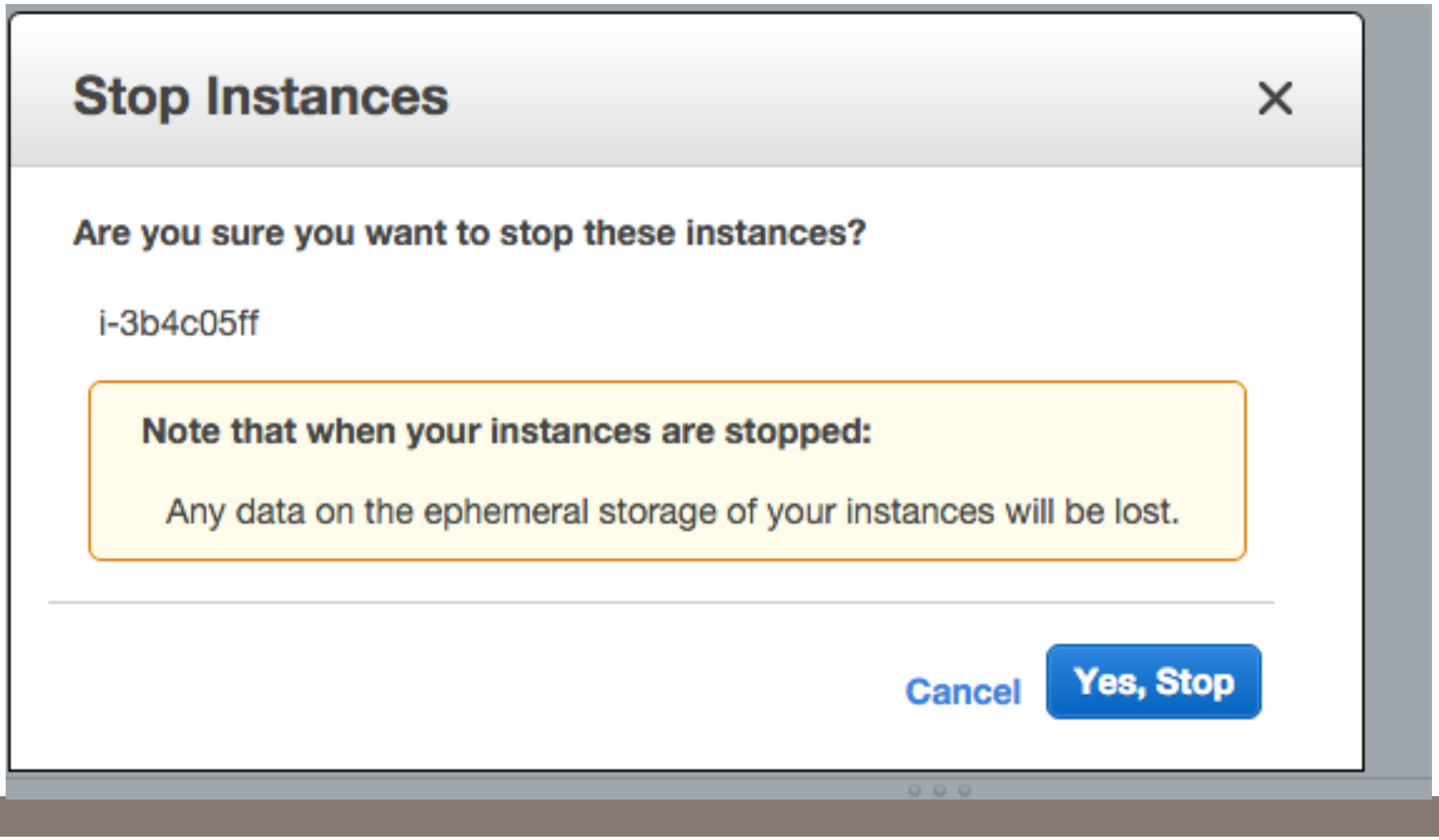
```
$ ssh -i ~/.ssh/anly502.pem ec2-user@52.33.99.98

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/
3 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
$
```

We have a running instance!

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.8G  1.1G  6.6G  15% /
devtmpfs        489M   56K  489M   1% /dev
tmpfs           498M     0  498M   0% /dev/shm
$
$ top
top - 21:46:19 up 40 min,  1 user,  load average: 0.03, 0.04, 0.03
Tasks:  68 total,   1 running,  67 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1019452k total,   326648k used,   692804k free,     9572k buffers
Swap:        0k total,        0k used,        0k free,   265376k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      20   0 19612 2536 2216 S  0.0  0.2   0:00.79 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.01 ksoftirqd/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:00.09 kworker/0:0
    5 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0     0    0    0 S  0.0  0.0   0:00.01 kworker/u30:0
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.03 rcu_sched
    8 root      20   0     0    0    0 S  0.0  0.0   0:00.00 rcu_bh
    9 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
   10 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 khelper
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kdevtmpfs
   12 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 netns
   13 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 perf
   14 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/u30:1
   16 root      20   0     0    0    0 S  0.0  0.0   0:00.01 xenwatch
   21 root      20   0     0    0    0 S  0.0  0.0   0:00.00 xenbus
```

Don’t forget to shut down when done!




Stop Instances

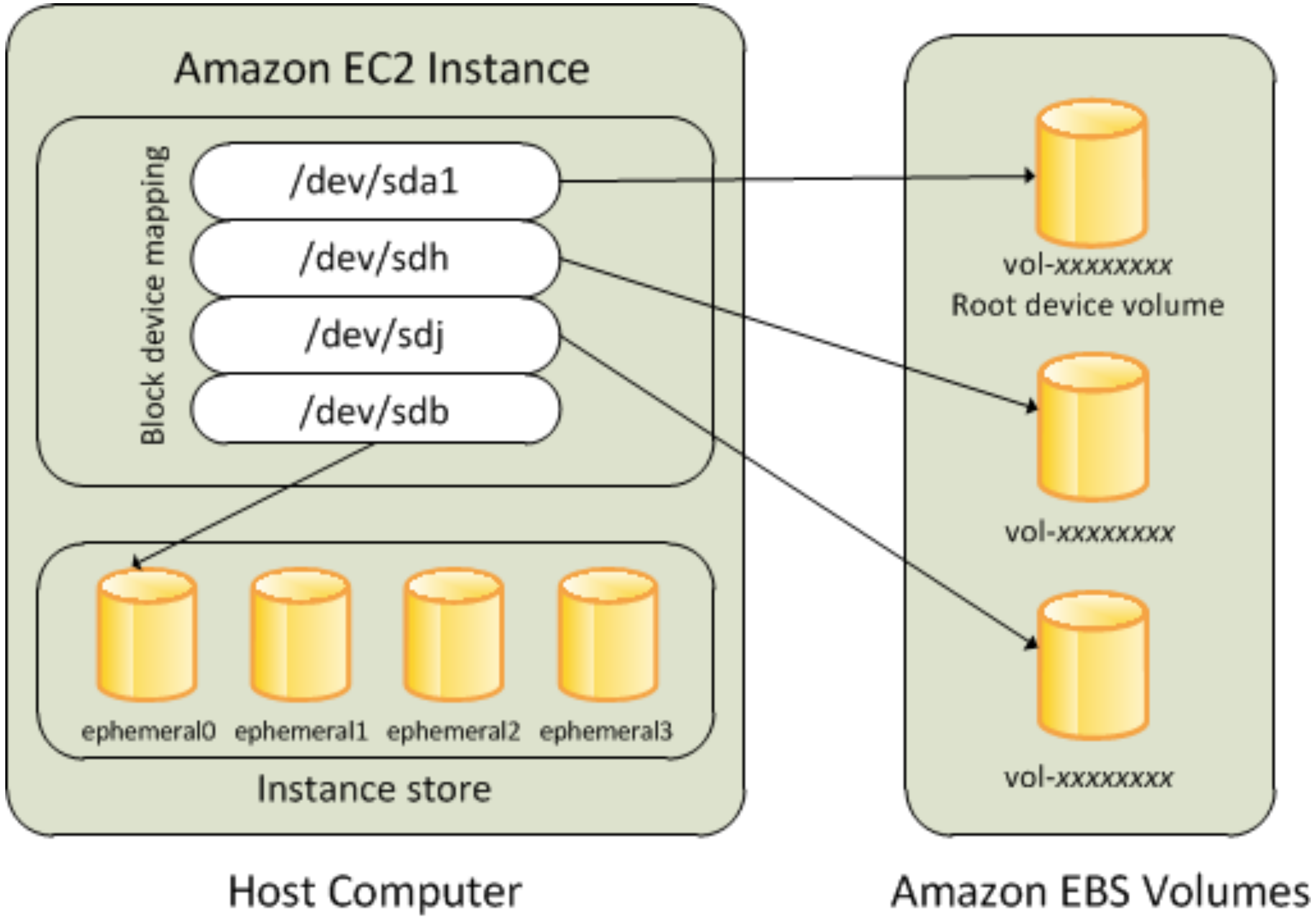
Are you sure you want to stop these instances?

i-3b4c05ff

Note that when your instances are stopped:

Any data on the ephemeral storage of your instances will be lost.

Ephemeral storage — part of the instance (local drives) faster.
EBS — separate devices — slower, but can persist.



http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/block-device-mapping-concepts.html

$$ DANGER $$

Remember — There are many different regions!

3
2
3
2
3
5
3
2
2
2
2
3
2
2
3
3

Global Infrastructure

# **Region & Number of Availability Zones**

AWS GovCloud (2)

US West
Oregon (3), Northern California (3)

US East
Northern Virginia (5), Ohio (3)

Canada
Central (2)

South America
São Paulo (3)

Europe
Ireland (3), Frankfurt (2), London (2)

Asia Pacific
Singapore (2), Sydney (3), Tokyo (3), Seoul (2), Mumbai (2)

China
Beijing (2)

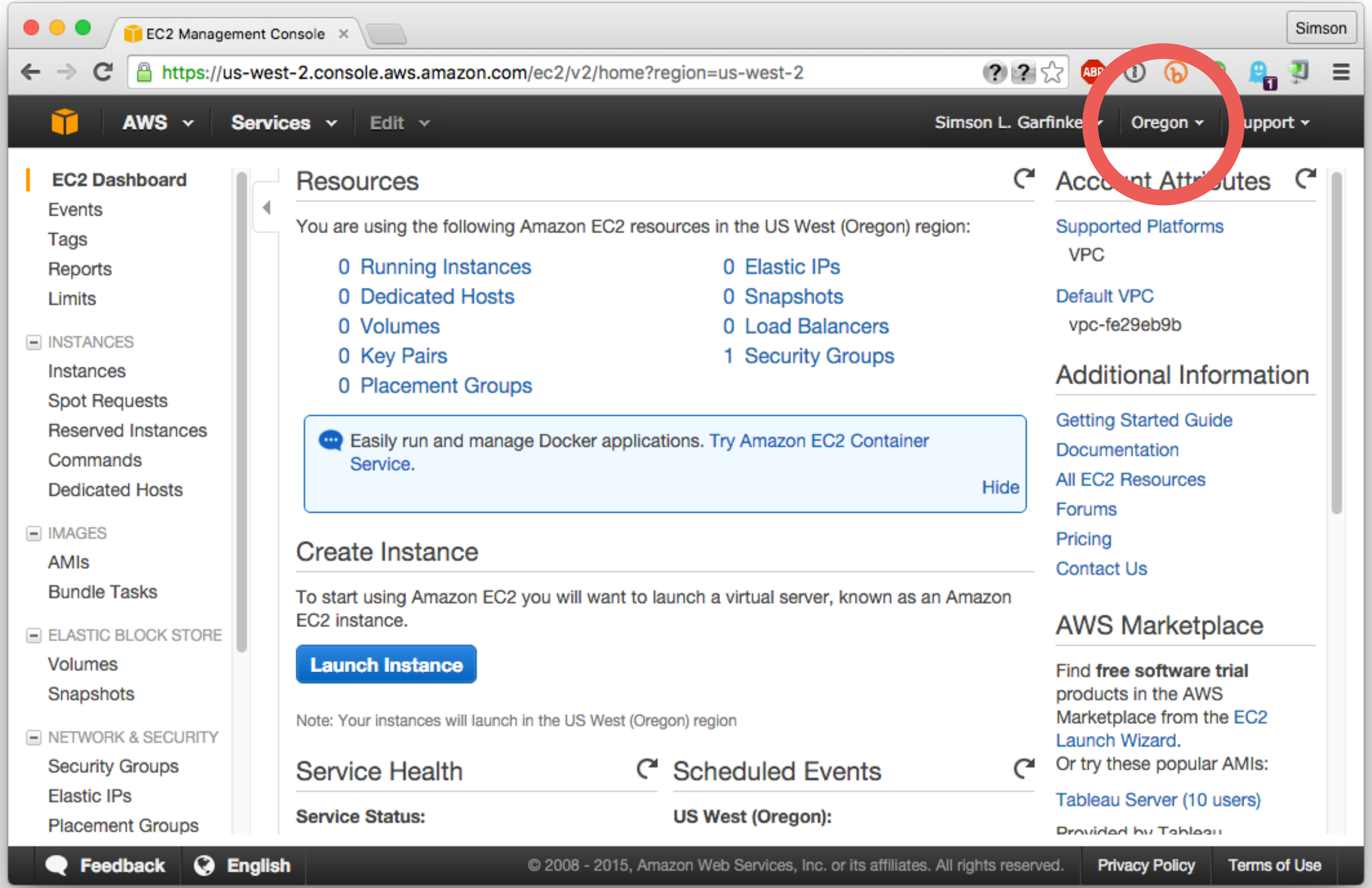
○ **New Regi**

Paris

Ningxia

https://aws.amazon.com/about-aws/global-infrastructure/

Keep watch on the "region"
You may have instances running elsewhere at Amazon…



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	

Easily run and manage Docker applications. Try Amazon EC2 Container Service.

Create Instance

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 West (Oregon) region

Service Health

Service Status:

Scheduled Events

US West (Oregon):

Account Attributes

Supported Platforms

VPC

Default VPC

vpc-fe29eb9b

Additional Information

Getting Started Guide

Documentation

All EC2 Resources

Forums

Pricing

Contact Us

AWS Marketplace

Find **free software trial** products in the AWS Marketplace from the EC2 Launch Wizard.

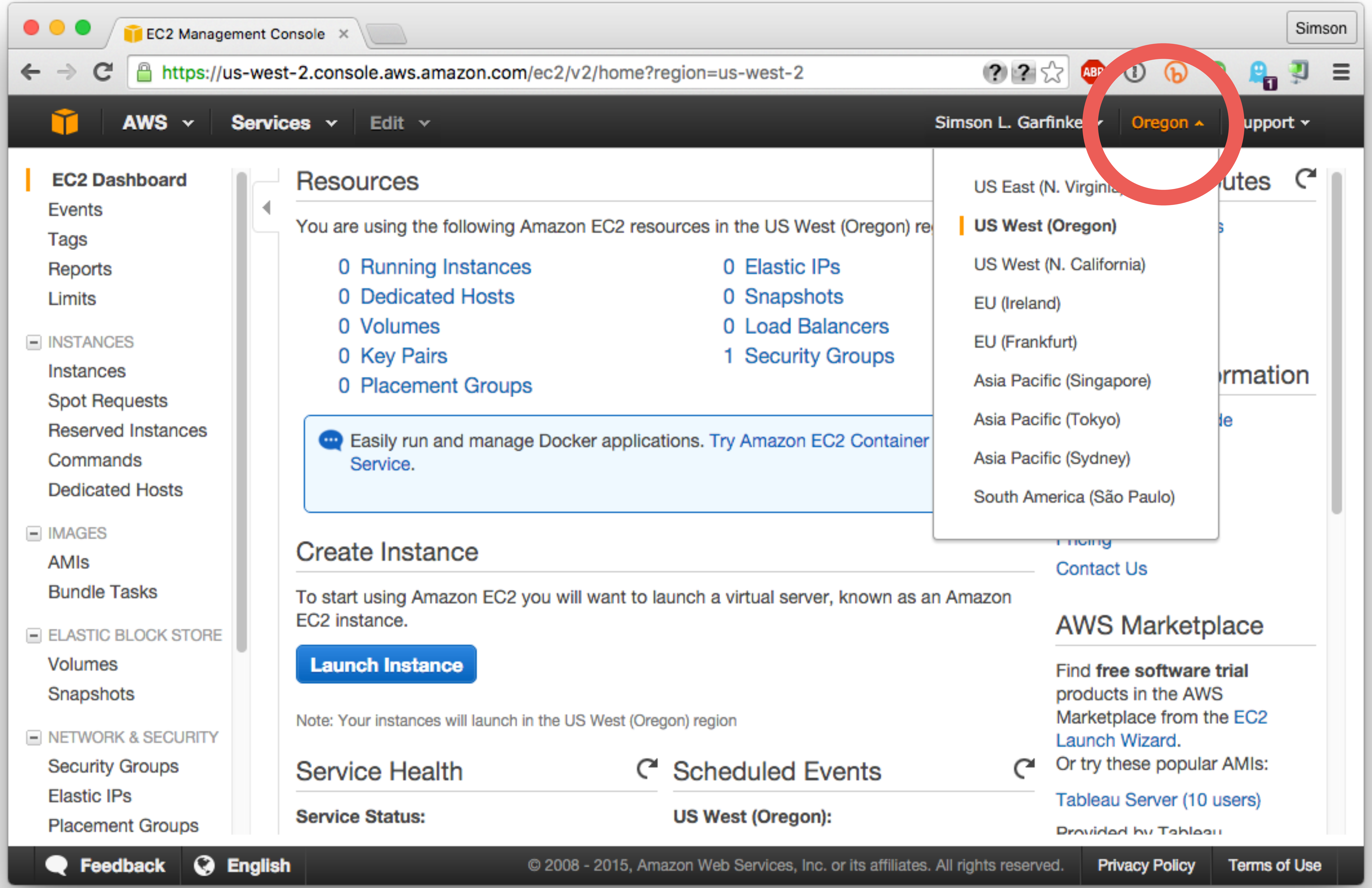
Or try these popular AMIs:

Tableau Server (10 users)

Provided by Tableau

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep watch on the "region"
You may have instances running elsewhere at Amazon…



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) re

0 Running Instances
0 Dedicated Hosts
0 Volumes
0 Key Pairs
0 Placement Groups

0 Elastic IPs
0 Snapshots
0 Load Balancers
1 Security Groups

Easily run and manage Docker applications. Try Amazon EC2 Container Service.

Create Instance

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 West (Oregon) region

Service Health

Service Status:

Scheduled Events

US West (Oregon):

Contact Us

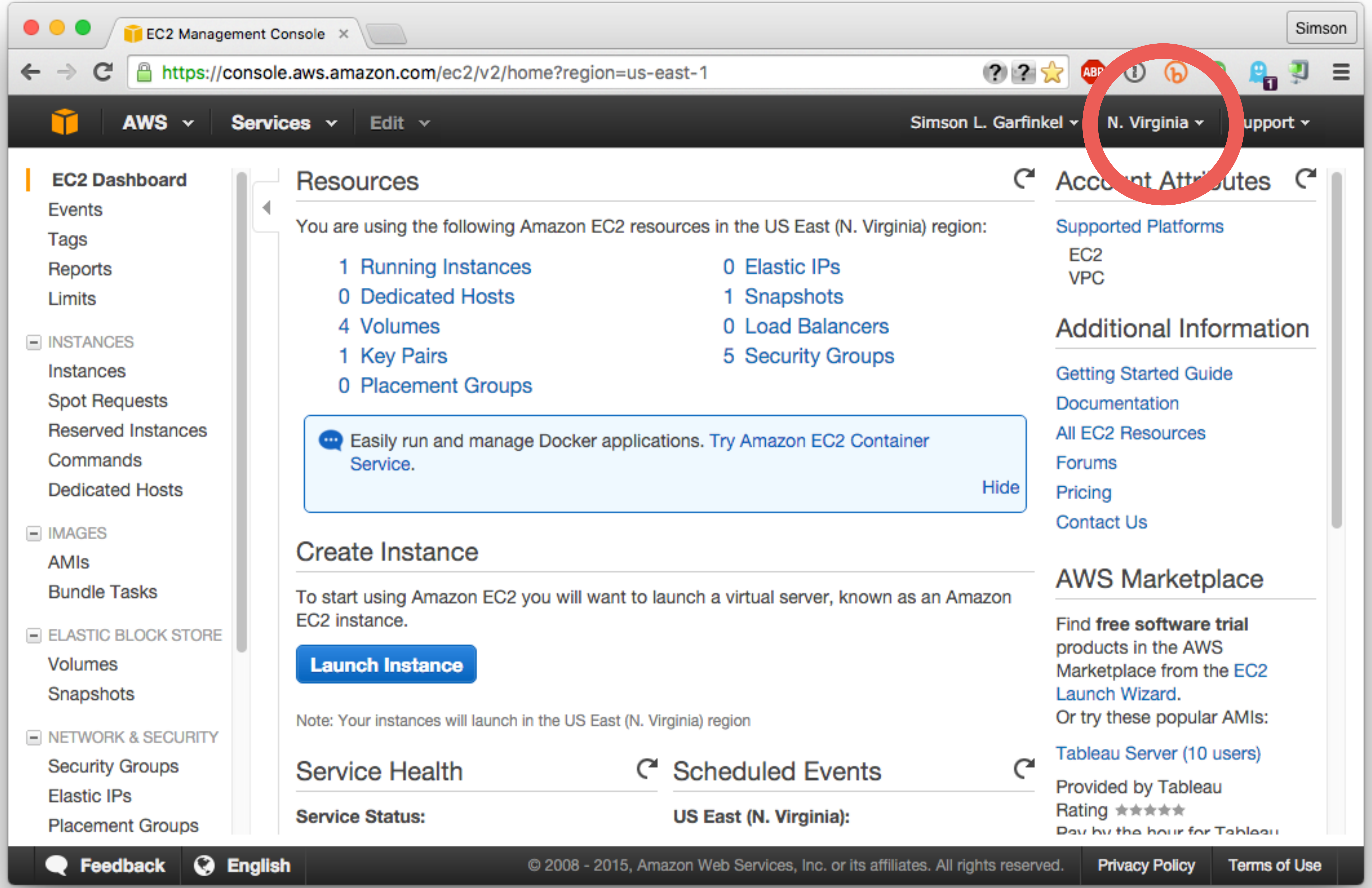
AWS Marketplace

Find **free software trial** products in the AWS Marketplace from the EC2 Launch Wizard.
Or try these popular AMIs:

Tableau Server (10 users)

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep watch on the "region"
You may have instances running elsewhere at Amazon…



Resources

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

1 Running Instances	0 Elastic IPs
0 Dedicated Hosts	1 Snapshots
4 Volumes	0 Load Balancers
1 Key Pairs	5 Security Groups
0 Placement Groups	

Easily run and manage Docker applications. Try Amazon EC2 Container Service.

Create Instance

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 (N. Virginia) region

Service Health

Service Status:

Scheduled Events

US East (N. Virginia):

Account Attributes

Supported Platforms

EC2
VPC

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

AWS Marketplace

Find **free software trial** products in the AWS Marketplace from the EC2 Launch Wizard.
Or try these popular AMIs:

Tableau Server (10 users)
Provided by Tableau
Rating ★★★★★
Pay by the hour for Tableau

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

LAST YEAR,

A STUDENT LOST **$250** LEAVING A CLUSTER RUNNING IN ANOTHER REGION!

If you need more RAM on EC2, you can always swap!

t2.micro instances have 1GB of physical RAM.

If you need more, but don’t want to create a bigger instance, you can swap.

Here’s how to create an 8GB swap file:

```
$ sudo bash
# dd if=/dev/zero of=/var/swapfile bs=1048576 count=4096
# chown 600 /var/swapfile
# mkswap /var/swapfile
# swapon /var/swapfile
# vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 4  0   4248  96328  29424 236608    0    0    11   126   20   32  2  0 98  0  0
#
```

Remember — swapping slows down a system significantly.

- “You can’t fake what you don’t have.” — Seymour Cray

EC2 Command Line Tools

```
$ aws ec2 describe-instances
```

Amazon provides command line tools

Can be run from *any* Linux, Mac or Windows computer.

- Faster interaction than web interface.
- Can be scripted.

AWS Command Line Interface

- Run through “aws” command
- Flexible output — JSON, text, tables
- List EC2 instance: $ aws ec2 describe-instances
    - https://aws.amazon.com/cli/
    - http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html

Elastic Comput Cloud CLI

- Run through 176 different ec2-* commands
- List EC2 instances: $ ec2-describe-instances
    - http://docs.aws.amazon.com/AWSEC2/latest/CommandLineReference/ApiReference-cmd-DescribeVolumes.html

Credentials:

- Credentials kept in $HOME/.aws/ directory
- Credentials kept in AWS_USERNAME, AWS_ACCESS_KEY, AWS_SECRET_KEY environment variables.

Both are pre-installed on Amazon’s AMIs. Use the AWS CLI if possible.

Set up your environment variables and test:

AWS CLI command:

```
$ aws ec2 describe-regions
REGIONS      ec2.eu-west-1.amazonaws.com eu-west-1
REGIONS      ec2.ap-southeast-1.amazonaws.com     ap-southeast-1
REGIONS      ec2.ap-southeast-2.amazonaws.com     ap-southeast-2
REGIONS      ec2.eu-central-1.amazonaws.com       eu-central-1
REGIONS      ec2.ap-northeast-1.amazonaws.com     ap-northeast-1
REGIONS      ec2.us-east-1.amazonaws.com us-east-1
REGIONS      ec2.sa-east-1.amazonaws.com sa-east-1
REGIONS      ec2.us-west-1.amazonaws.com us-west-1
REGIONS      ec2.us-west-2.amazonaws.com us-west-2
```

(Old-style EC2- command)

```
$ ec2-describe-regions
REGION       eu-west-1 ec2.eu-west-1.amazonaws.com
REGION       ap-southeast-1     ec2.ap-southeast-1.amazonaws.com
REGION       ap-southeast-2     ec2.ap-southeast-2.amazonaws.com
REGION       eu-central-1       ec2.eu-central-1.amazonaws.com
REGION       ap-northeast-1     ec2.ap-northeast-1.amazonaws.com
REGION       us-east-1 ec2.us-east-1.amazonaws.com
REGION       sa-east-1 ec2.sa-east-1.amazonaws.com
REGION       us-west-1 ec2.us-west-1.amazonaws.com
REGION       us-west-2 ec2.us-west-2.amazonaws.com
```

EC2 has a command-line interface

Show running instances:

```
$ aws ec2 describe-instances --output text
RESERVATION              r-d9792009       376778049323
INSTANCEi-5c306beb       ami-60b6c60a                      ip-172-30-1-33.ec2.internal       running          mucha            0                                 t2.micro         2015-12-02T01:47:21+0000
        us-east-1b                                                            monitoring-disabled               52.90.221.164    172.30.1.33      vpc-8e73cfeb     subnet-b1de03c6  ebs
                                          hvm              xen              haLAd1447616090330                sg-15edc370      default          false            arn:aws:iam::376778049323:instance-profile/
MyWebApplication
BLOCKDEVICE              /dev/xvda        vol-8f73a76c     2015-11-15T19:34:54.000Z          true
NIC     eni-7cd9c431     subnet-b1de03c6  vpc-8e73cfeb     376778049323     in-use           172.30.1.33                       true
NICATTACHMENT            eni-attach-9d6cc676               0                attached         2015-11-15T14:34:50-0500          true
NICASSOCIATION           52.90.221.164    amazon           172.30.1.33
GROUP   sg-15edc370      default
PRIVATEIPADDRESS         172.30.1.33
TAG     instance         i-5c306beb       Name             Persistent EC2
RESERVATION              r-d05e523a       376778049323
INSTANCEi-eba98616       ami-1ecae776                      ip-172-30-1-209.ec2.internal      stopped          mucha            0                                 t2.micro         2015-05-08T21:06:32+0000
        us-east-1b                                                            monitoring-disabled                                172.30.1.209     vpc-8e73cfeb     subnet-b1de03c6  ebs
                                          hvm              xen              ESeOf1431119191963                sg-15edc370      default          false
BLOCKDEVICE              /dev/xvda        vol-8ff91561     2015-05-08T21:06:37.000Z          true
NIC     eni-d79fdb9f     subnet-b1de03c6  vpc-8e73cfeb     376778049323     in-use           172.30.1.209                      true
NICATTACHMENT            eni-attach-6dd55108               0                attached         2015-05-08T17:06:32-0400          true
GROUP   sg-15edc370      default
PRIVATEIPADDRESS         172.30.1.209
TAG     instance         i-eba98616       Name             TLSA Tester
RESERVATION              r-00e7e5d0       376778049323
INSTANCEi-9a48aa2c       ami-d05e75b8                      ip-172-30-1-247.ec2.internal      running          mucha            0                                 t2.micro         2015-12-04T13:15:04+0000
        us-east-1b                                                            monitoring-disabled               54.85.124.24     172.30.1.247     vpc-8e73cfeb     subnet-b1de03c6  ebs
                                          hvm              xen              TlrLB1448489248412                sg-e8e5ad8e      default          false            arn:aws:iam::376778049323:instance-profile/
MyWebApplication
BLOCKDEVICE              /dev/sda1        vol-e8e16e0b     2015-11-25T22:07:31.000Z          false
NIC     eni-30b8687c     subnet-b1de03c6  vpc-8e73cfeb     376778049323     in-use           172.30.1.247                      true
NICATTACHMENT            eni-attach-f5bfd91e               0                attached         2015-11-25T17:07:29-0500          true
NICASSOCIATION           54.85.124.24     amazon           172.30.1.247
GROUP   sg-e8e5ad8e      residual-study
PRIVATEIPADDRESS         172.30.1.247
PRIVATEIPADDRESS         172.30.1.35
TAG     instance         i-9a48aa2c       Name             Reminance Study
RESERVATION              r-ba2ffc90       376778049323
INSTANCEi-8e0b7f64       ami-ba13abd2                      ip-172-30-1-89.ec2.internal       stopped          windows1         0                                 t2.micro         2015-04-26T14:40:59+0000
        us-east-1b                                         windows          monitoring-disabled                                172.30.1.89      vpc-8e73cfeb     subnet-b1de03c6  ebs
                                          hvm              xen              PEYEX1416103617266                sg-15edc370      default          false
BLOCKDEVICE              /dev/sda1        vol-65202e2d     2014-11-16T02:07:01.000Z          true
NIC     eni-9f546be9     subnet-b1de03c6  vpc-8e73cfeb     376778049323     in-use           172.30.1.89                       true
NICATTACHMENT            eni-attach-30898753               0                attached         2014-11-15T21:06:57-0500          true
GROUP   sg-15edc370      default
PRIVATEIPADDRESS         172.30.1.89
TAG     instance         i-8e0b7f64       Name             Quicken
[Dance ~ 10:34:10]$
```

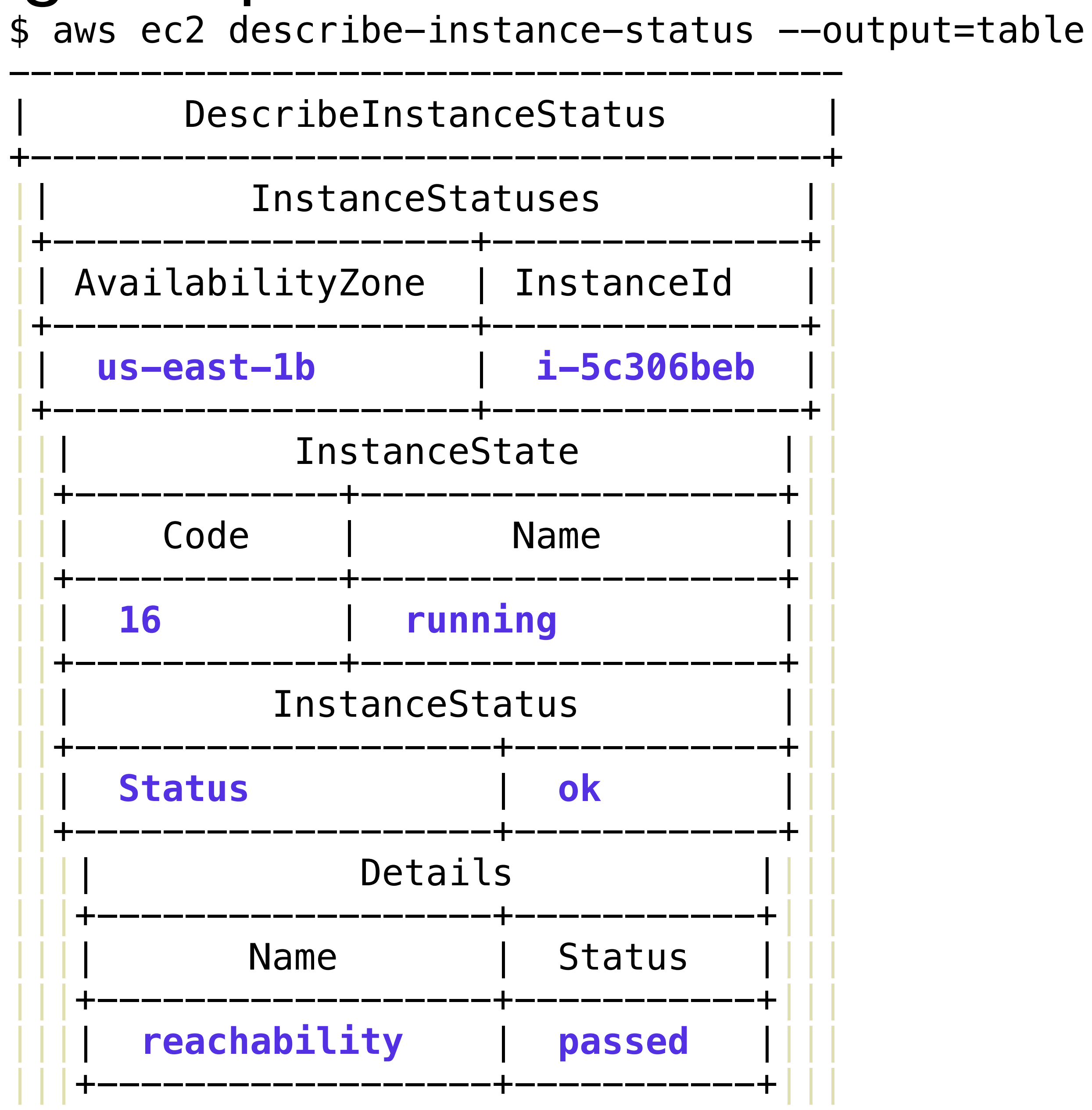
Use “help” to get help

```
$ aws ec2 describe-instances help


NAME
       describe-instances -

DESCRIPTION
       Describes one or more of your instances.

       If you specify one or more instance IDs, Amazon EC2 returns information
       for those instances. If you do not specify  instance  IDs,  Amazon  EC2
       returns  information  for  all  relevant  instances.  If you specify an
       instance ID that is not valid, an error is returned. If you specify  an
       instance  that  you  do  not  own,  it  is not included in the returned
       results.

       Recently terminated instances might appear  in  the  returned  results.
       This interval is usually less than one hour.

       describe-instances  is a paginated operation. Multiple API calls may be
       issued in order to retrieve the entire data set  of  results.  You  can
       disable pagination by providing the --no-paginate argument.  When using
       --output text and the --query argument on  a  paginated  response,  the
       --query  argument  must  extract data from the results of the following
       query expressions: Reservations

SYNOPSIS
            describe-instances
          [--dry-run | --no-dry-run]
          [--instance-ids <value>]
          [--filters <value>]
          [--cli-input-json <value>]
          [--starting-token <value>]
          [--page-size <value>]
          [--max-items <value>]
          [--generate-cli-skeleton]
```

$ ec2-describe-instance-status — see what's running

```
$ aws ec2 describe-instance-status --output=text
INSTANCESTATUSES    us-east-1b    i-5c306beb
INSTANCESTATE       16            running
INSTANCESTATUS      ok
DETAILS             reachability  passed
SYSTEMSTATUS        ok
DETAILS             reachability  passed
INSTANCESTATUSES    us-east-1b    i-9a48aa2c
INSTANCESTATE       16            running
INSTANCESTATUS      ok
DETAILS             reachability  passed
SYSTEMSTATUS        ok
DETAILS             reachability  passed
$
```

Change output format:

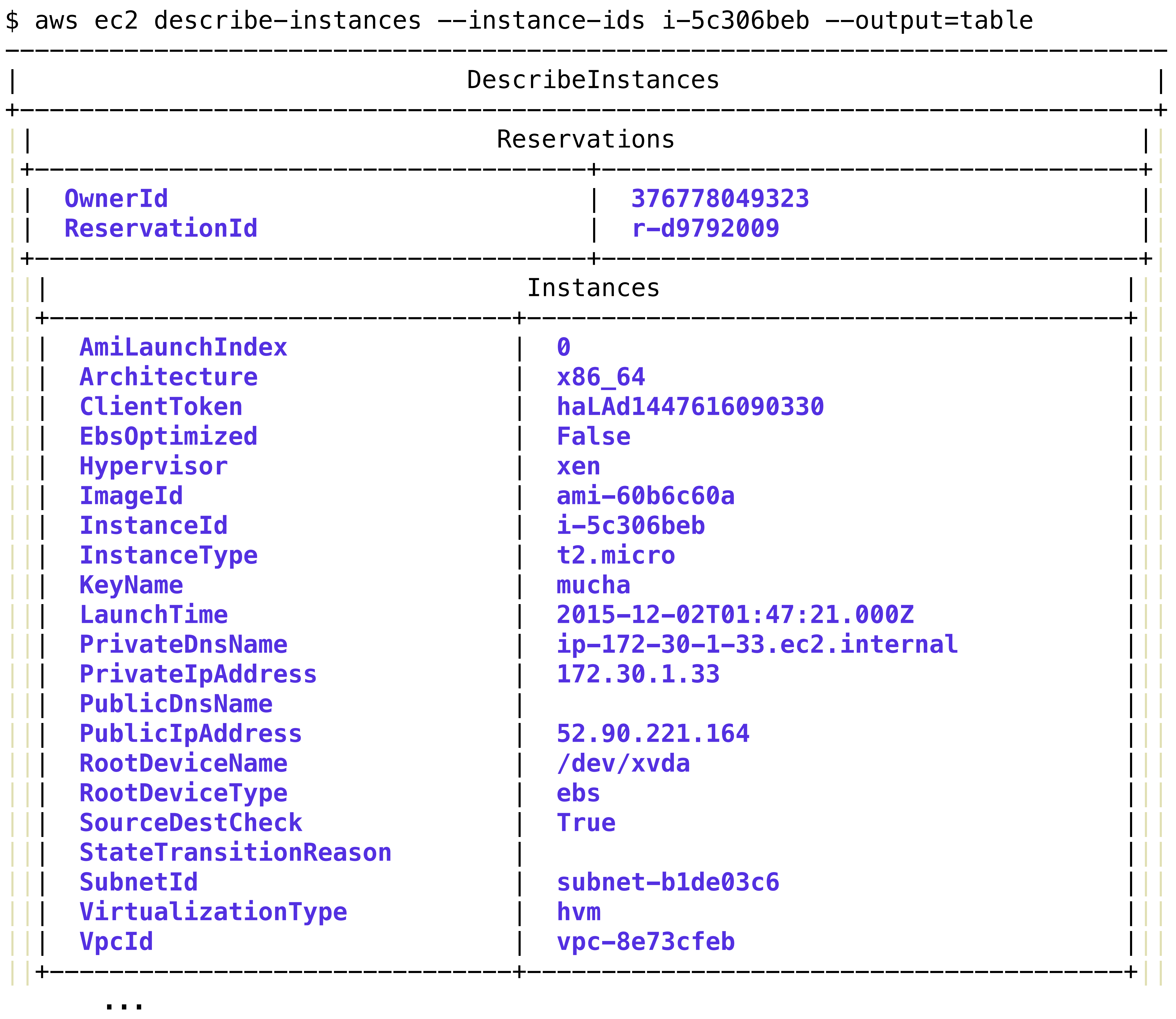


```
$ aws ec2 describe-instance-status --output=table
--------------------------------------
|       DescribeInstanceStatus       |
+------------------------------------+
||         InstanceStatuses         ||
|+-------------------+--------------+|
|| AvailabilityZone  | InstanceId   ||
|+-------------------+--------------+|
||  us-east-1b       |  i-5c306beb  ||
|+-------------------+--------------+|
|||          InstanceState          |||
||+------------+-------------------+||
|||    Code    |       Name        |||
||+------------+-------------------+||
|||  16        |  running          |||
||+------------+-------------------+||
|||         InstanceStatus          |||
||+------------------+--------------+||
|||  Status          |  ok          |||
||+------------------+--------------+||
||||           Details            ||||
|||+-----------------+------------+|||
||||       Name      |  Status    ||||
|||+-----------------+------------+|||
||||  reachability   |  passed    ||||
|||+-----------------+------------+|||
```

JSON output is more useful for scripting

```
$ aws ec2 describe-instance-status --output=json
{
    "InstanceStatuses": [
        {
            "InstanceId": "i-5c306beb",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
            "AvailabilityZone": "us-east-1b",
            "SystemStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            },
            "InstanceStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            }
        },
        {
            "InstanceId": "i-9a48aa2c",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
...
```

$ aws ec2 describe-instances —instance-ids=instance-id



```
$ aws ec2 describe-instances --instance-ids i-5c306beb --output=table
-------------------------------------------------------------------------------
|                                   DescribeInstances                         |
+-----------------------------------------------------------------------------+
||                               Reservations                                ||
|+--------------------------------------+------------------------------------+|
||  OwnerId                             |  376778049323                      ||
||  ReservationId                       |  r-d9792009                        ||
|+--------------------------------------+------------------------------------+|
|||                                 Instances                               |||
||+--------------------------------+----------------------------------------+||
|||  AmiLaunchIndex                |  0                                     |||
|||  Architecture                  |  x86_64                                |||
|||  ClientToken                   |  haLAd1447616090330                    |||
|||  EbsOptimized                  |  False                                 |||
|||  Hypervisor                    |  xen                                   |||
|||  ImageId                       |  ami-60b6c60a                          |||
|||  InstanceId                    |  i-5c306beb                            |||
|||  InstanceType                  |  t2.micro                              |||
|||  KeyName                       |  mucha                                 |||
|||  LaunchTime                    |  2015-12-02T01:47:21.000Z              |||
|||  PrivateDnsName                |  ip-172-30-1-33.ec2.internal           |||
|||  PrivateIpAddress              |  172.30.1.33                           |||
|||  PublicDnsName                 |                                        |||
|||  PublicIpAddress               |  52.90.221.164                         |||
|||  RootDeviceName                |  /dev/xvda                             |||
|||  RootDeviceType                |  ebs                                   |||
|||  SourceDestCheck               |  True                                  |||
|||  StateTransitionReason         |                                        |||
|||  SubnetId                      |  subnet-b1de03c6                       |||
|||  VirtualizationType            |  hvm                                   |||
|||  VpcId                         |  vpc-8e73cfeb                          |||
||+--------------------------------+----------------------------------------+||
       ...
```

Per-instance metadata:
Letting the instance know what it is

HTTP API:

```
$ curl http://169.254.169.254/latest/meta-data/instance-id
i-5c306beb$

$ aws_instance=$(wget -q -O- http://169.254.169.254/latest/meta-data/instance-id)
$ aws_region=$(wget -q -O- http://169.254.169.254/latest/meta-data/hostname)
$ echo $aws_instance $aws_region
i-5c306beb ip-172-30-1-33.ec2.internal
$
```

- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html

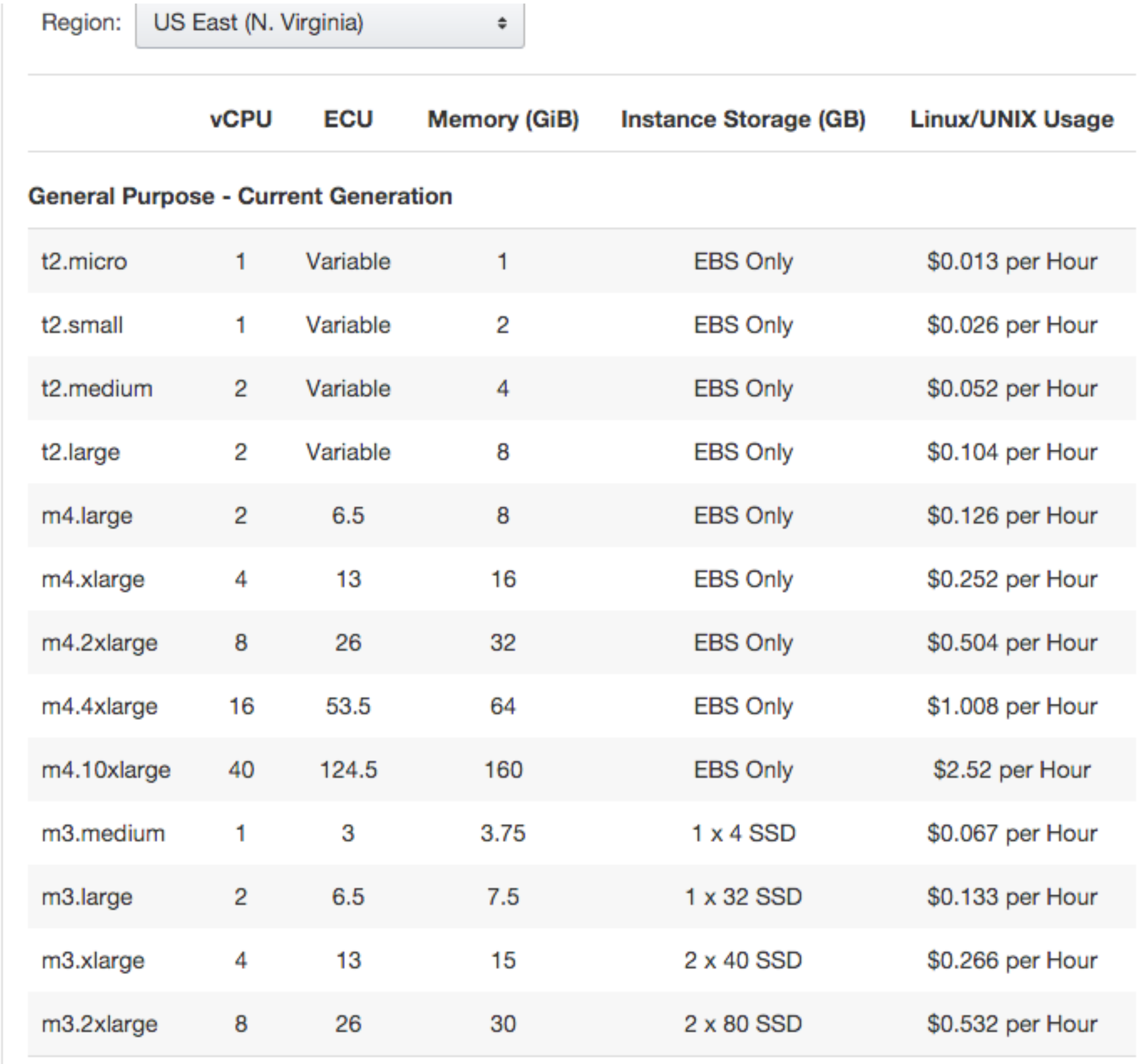
ec2-metadata:

```
$ ec2-metadata  -i
instance-id: i-5c306beb
$ ec2-metadata  -i| awk '{print $2;}'
i-5c306beb
```

EC2 Pricing

$$

Current EC2 pricing…



Region: US East (N. Virginia)

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
t2.micro	1	Variable	1	EBS Only	$0.013 per Hour
t2.small	1	Variable	2	EBS Only	$0.026 per Hour
t2.medium	2	Variable	4	EBS Only	$0.052 per Hour
t2.large	2	Variable	8	EBS Only	$0.104 per Hour
m4.large	2	6.5	8	EBS Only	$0.126 per Hour
m4.xlarge	4	13	16	EBS Only	$0.252 per Hour
m4.2xlarge	8	26	32	EBS Only	$0.504 per Hour
m4.4xlarge	16	53.5	64	EBS Only	$1.008 per Hour
m4.10xlarge	40	124.5	160	EBS Only	$2.52 per Hour
m3.medium	1	3	3.75	1 x 4 SSD	$0.067 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	$0.133 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	$0.266 per Hour
m3.2xlarge	8	26	30	2 x 80 SSD	$0.532 per Hour

Costing your instance…

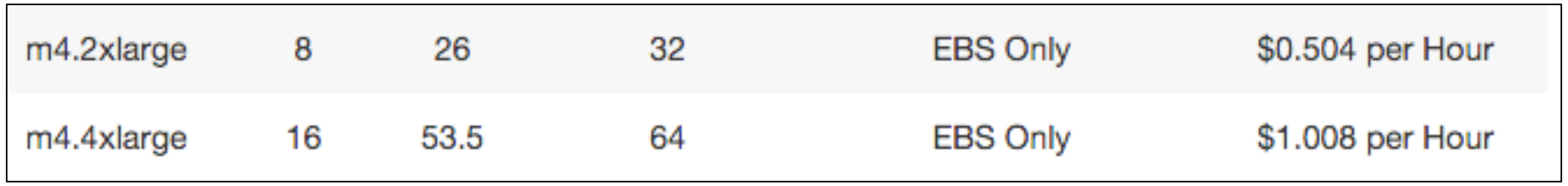
Amazon bills AMI’s at cost per hour…

… but you need to think about performance per dollar.

Performance is determined by:

- Location
- Total Memory
- Memory per core
- Storage
- Network
- Operating System

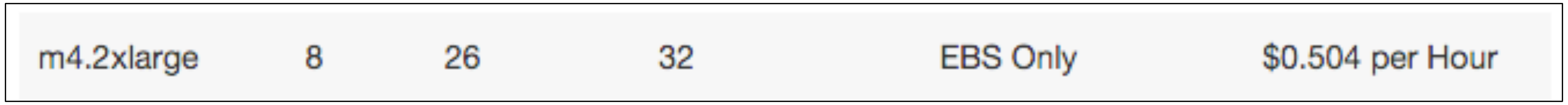
Will you get more work done with one m4.2xlarge or 2 m4.4xlarge?



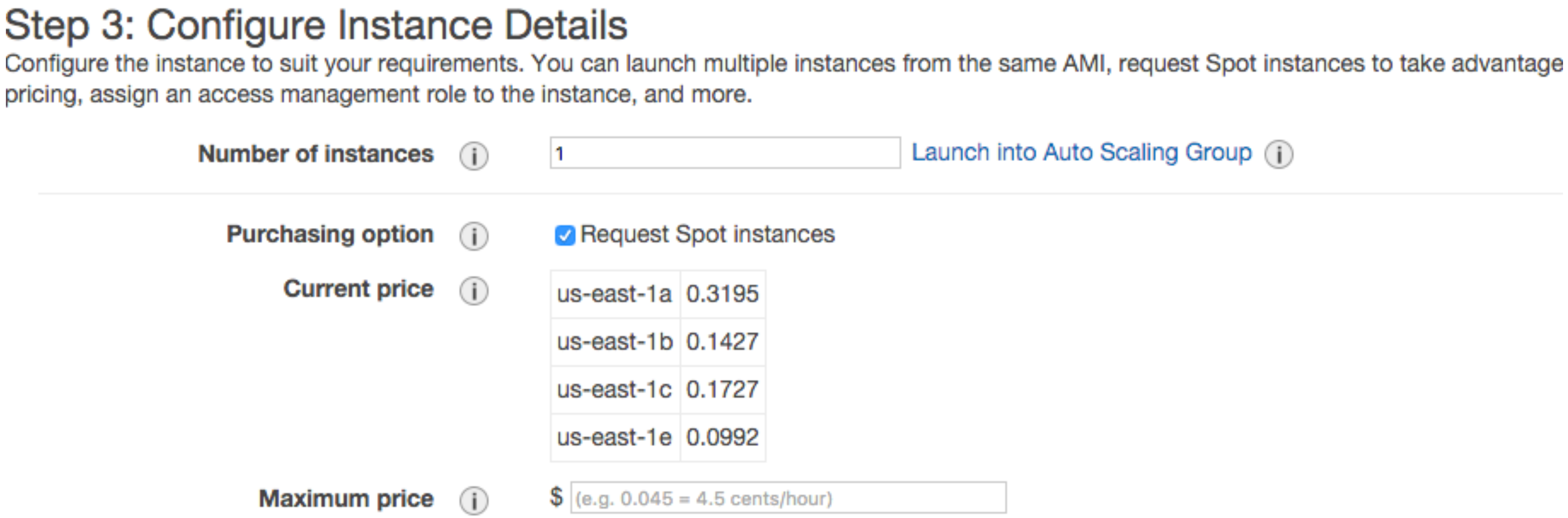
m4.2xlarge	8	26	32	EBS Only	$0.504 per Hour
m4.4xlarge	16	53.5	64	EBS Only	$1.008 per Hour

You can save a lot of money with “spot instances”

Fixed vs. Spot pricing, Nov 29, 2015:



m4.2xlarge	8	26	32	EBS Only	$0.504 per Hour



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage pricing, assign an access management role to the instance, and more.

Number of instances ⓘ 1 Launch into Auto Scaling Group ⓘ

Purchasing option ⓘ ☑ Request Spot instances

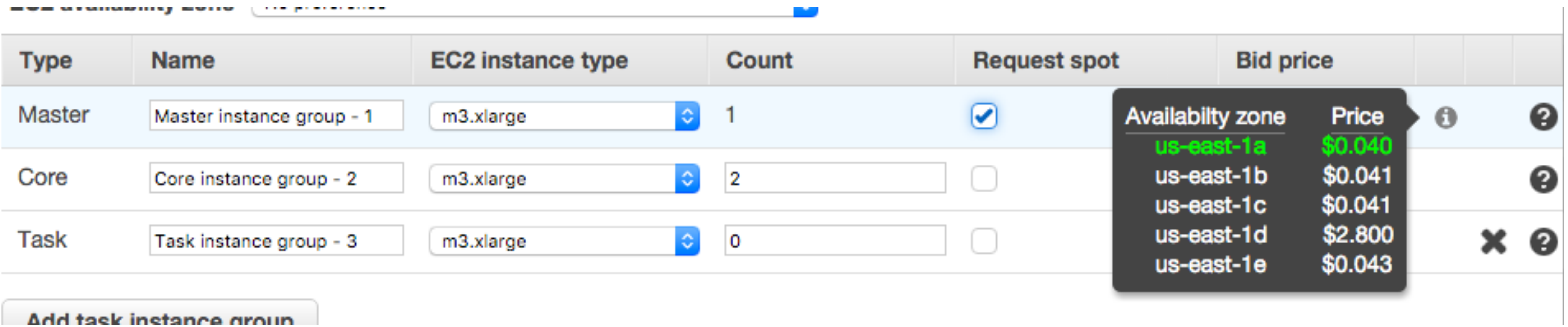
Current price ⓘ

us-east-1a	0.3195
us-east-1b	0.1427
us-east-1c	0.1727
us-east-1e	0.0992

Maximum price ⓘ $ (e.g. 0.045 = 4.5 cents/hour)

Even better spot prices!

Sunday, Feb 7, 2016:



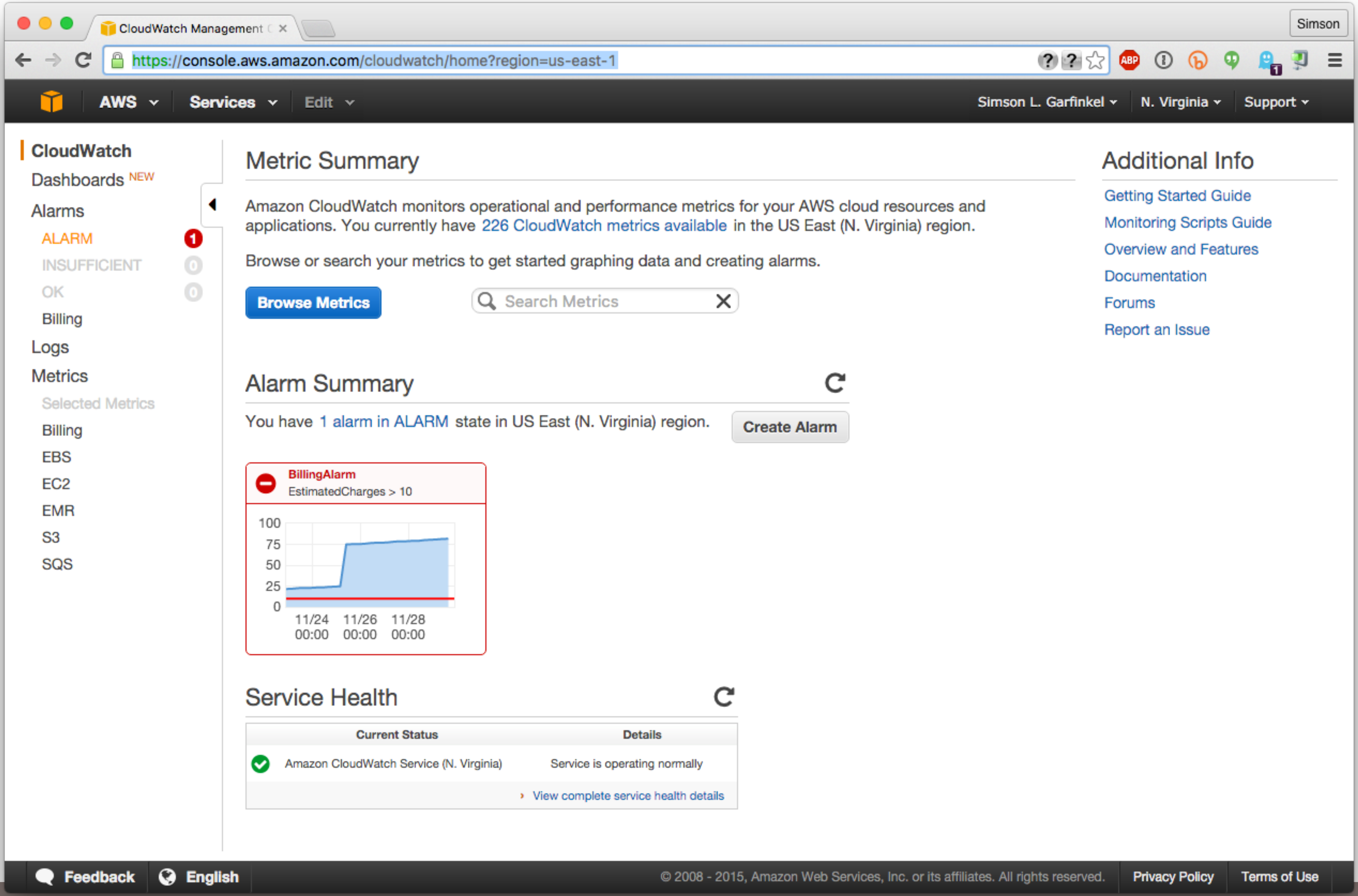
EC2 availability zone No preference

Type	Name	EC2 instance type	Count	Request spot	Bid price
Master	Master instance group - 1	m3.xlarge	1	☑	
Core	Core instance group - 2	m3.xlarge	2	□	
Task	Task instance group - 3	m3.xlarge	0	□	

Availabilty zone	Price
us-east-1a	$0.040
us-east-1b	$0.041
us-east-1c	$0.041
us-east-1d	$2.800
us-east-1e	$0.043

Amazon CloudWatch

CloudWatch alerts you if something is getting out of control.



Metric Summary

Amazon CloudWatch monitors operational and performance metrics for your AWS cloud resources and applications. You currently have 226 CloudWatch metrics available in the US East (N. Virginia) region.

Browse or search your metrics to get started graphing data and creating alarms.

Additional Info

Alarm Summary

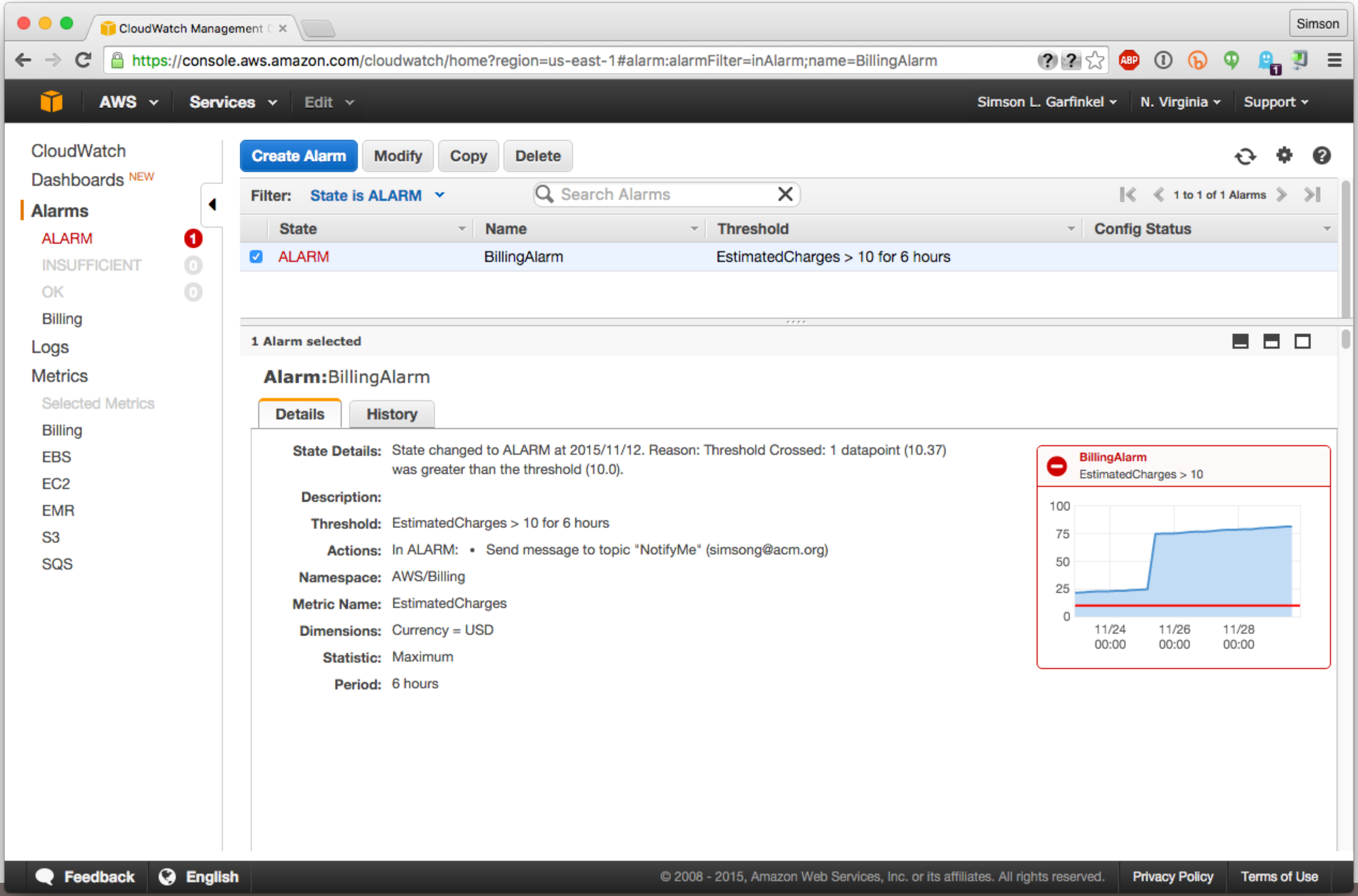
You have 1 alarm in ALARM state in US East (N. Virginia) region.

Service Health

Current Status	Details
✓ Amazon CloudWatch Service (N. Virginia)	Service is operating normally

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You should set up alarms!



Filter: State is ALARM

	State	Name	Threshold	Config Status
☑	ALARM	BillingAlarm	EstimatedCharges > 10 for 6 hours	

1 Alarm selected

Alarm: BillingAlarm

State Details: State changed to ALARM at 2015/11/12. Reason: Threshold Crossed: 1 datapoint (10.37) was greater than the threshold (10.0).

Description:

Threshold: EstimatedCharges > 10 for 6 hours

Actions: In ALARM: • Send message to topic "NotifyMe" (simsong@acm.org)

Namespace: AWS/Billing

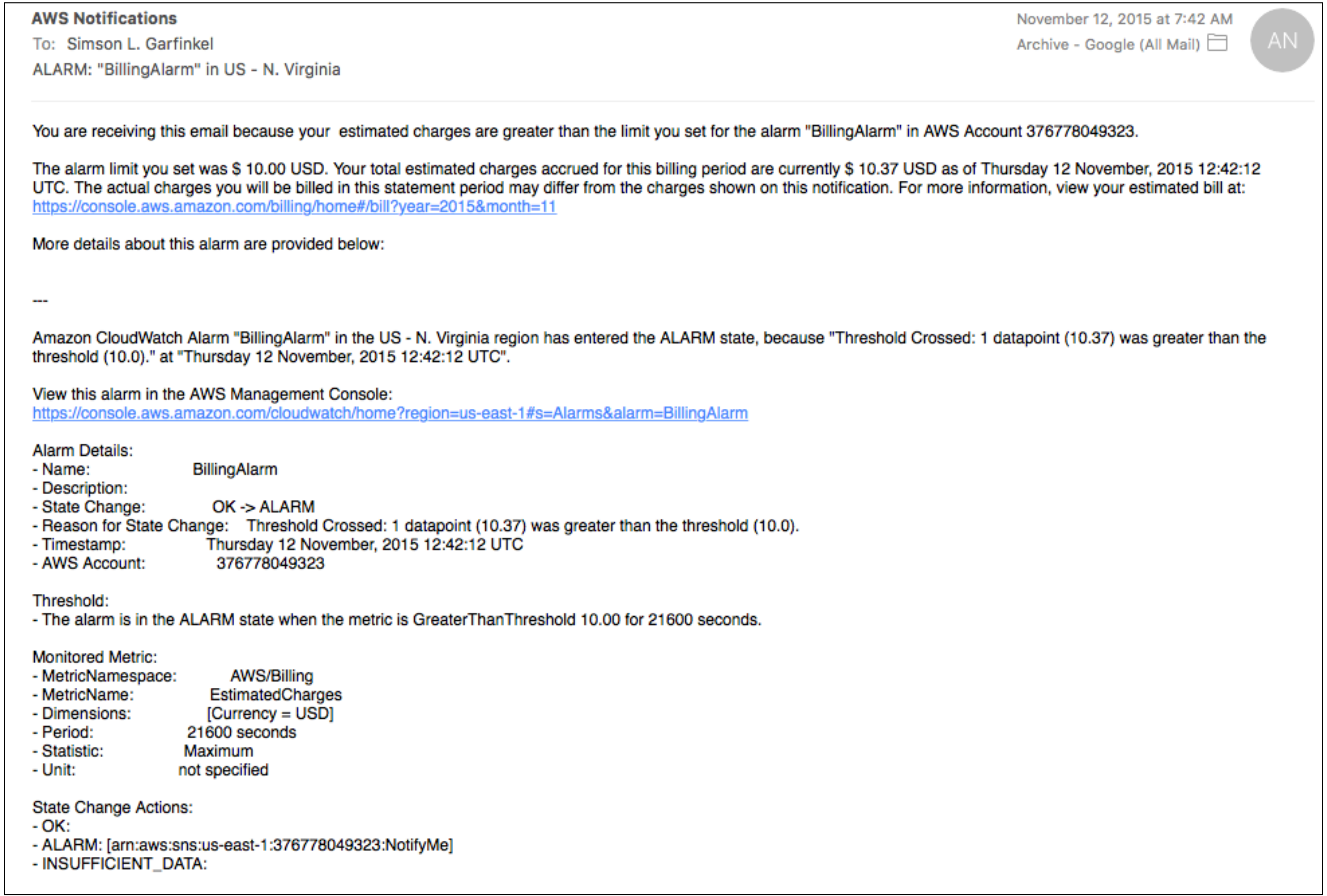
Metric Name: EstimatedCharges

Dimensions: Currency = USD

Statistic: Maximum

Period: 6 hours

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.



AWS Notifications
To: Simson L. Garfinkel
ALARM: "BillingAlarm" in US - N. Virginia

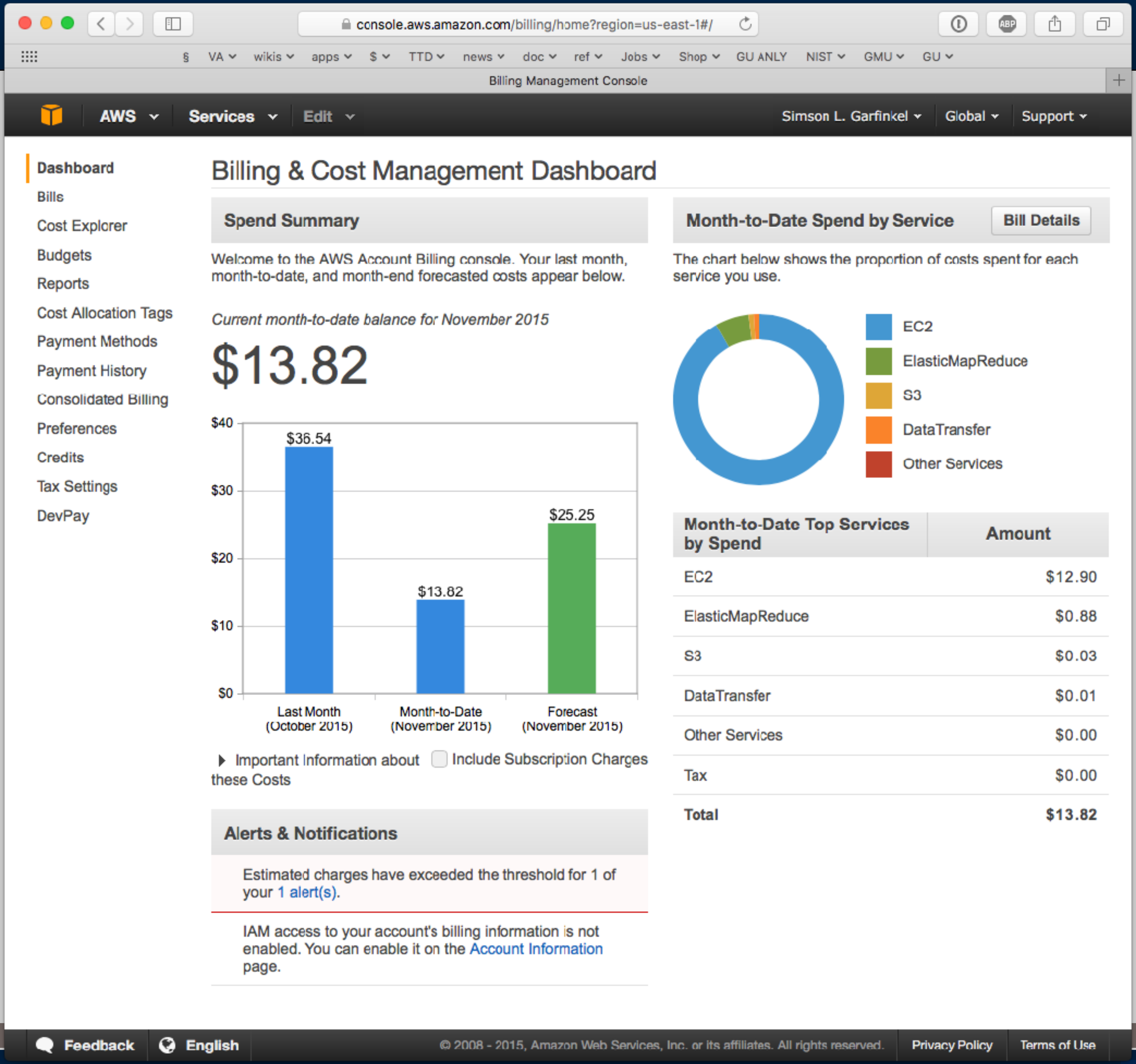
November 12, 2015 at 7:42 AM
Archive - Google (All Mail)

You are receiving this email because your estimated charges are greater than the limit you set for the alarm "BillingAlarm" in AWS Account 376778049323.

The alarm limit you set was $ 10.00 USD. Your total estimated charges accrued for this billing period are currently $ 10.37 USD as of Thursday 12 November, 2015 12:42:12 UTC. The actual charges you will be billed in this statement period may differ from the charges shown on this notification. For more information, view your estimated bill at:
https://console.aws.amazon.com/billing/home#/bill?year=2015&month=11

More details about this alarm are provided below:

```
---

Amazon CloudWatch Alarm "BillingAlarm" in the US - N. Virginia region has entered the ALARM state, because "Threshold Crossed: 1 datapoint (10.37) was greater than the threshold (10.0)." at "Thursday 12 November, 2015 12:42:12 UTC".

View this alarm in the AWS Management Console:
https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#s=Alarms&alarm=BillingAlarm

Alarm Details:
- Name:                      BillingAlarm
- Description:
- State Change:              OK -> ALARM
- Reason for State Change:   Threshold Crossed: 1 datapoint (10.37) was greater than the threshold (10.0).
- Timestamp:                 Thursday 12 November, 2015 12:42:12 UTC
- AWS Account:               376778049323

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 10.00 for 21600 seconds.

Monitored Metric:
- MetricNamespace:           AWS/Billing
- MetricName:                EstimatedCharges
- Dimensions:                [Currency = USD]
- Period:                    21600 seconds
- Statistic:                 Maximum
- Unit:                      not specified

State Change Actions:
- OK:
- ALARM: [arn:aws:sns:us-east-1:376778049323:NotifyMe]
- INSUFFICIENT_DATA:
```



Billing & Cost Management Dashboard

Spend Summary

Welcome to the AWS Account Billing console. Your last month, month-to-date, and month-end forecasted costs appear below.

Current month-to-date balance for November 2015

$13.82

▸ Important Information about these Costs

□ Include Subscription Charges

Month-to-Date Spend by Service

The chart below shows the proportion of costs spent for each service you use.

Month-to-Date Top Services by Spend	Amount
EC2	$12.90
ElasticMapReduce	$0.88
S3	$0.03
DataTransfer	$0.01
Other Services	$0.00
Tax	$0.00
Total	**$13.82**

Alerts & Notifications

Estimated charges have exceeded the threshold for 1 of your 1 alert(s).

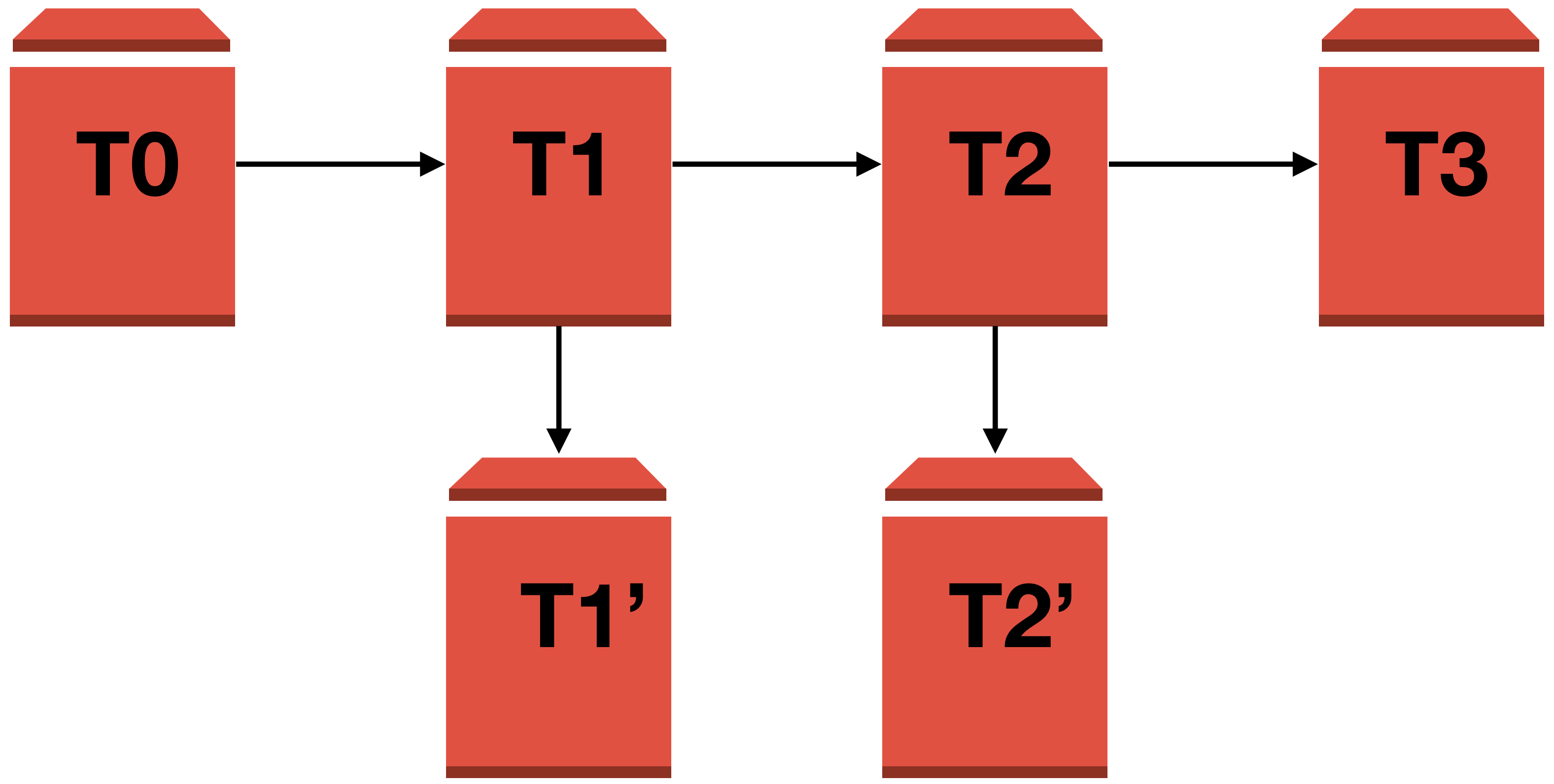
IAM access to your account's billing information is not enabled. You can enable it on the Account Information page.

Amazon EBS

EBS: Virtual Disk Volumes

EBS volumes:

- Created automatically when EC2 instance starts up.
- Snapshots on the fly.



Options:

- Magnetic or SSD
- Destroy or persist on instance termination
- Not Encrypted / Encrypted
- Provisioned IOPS

Uses:

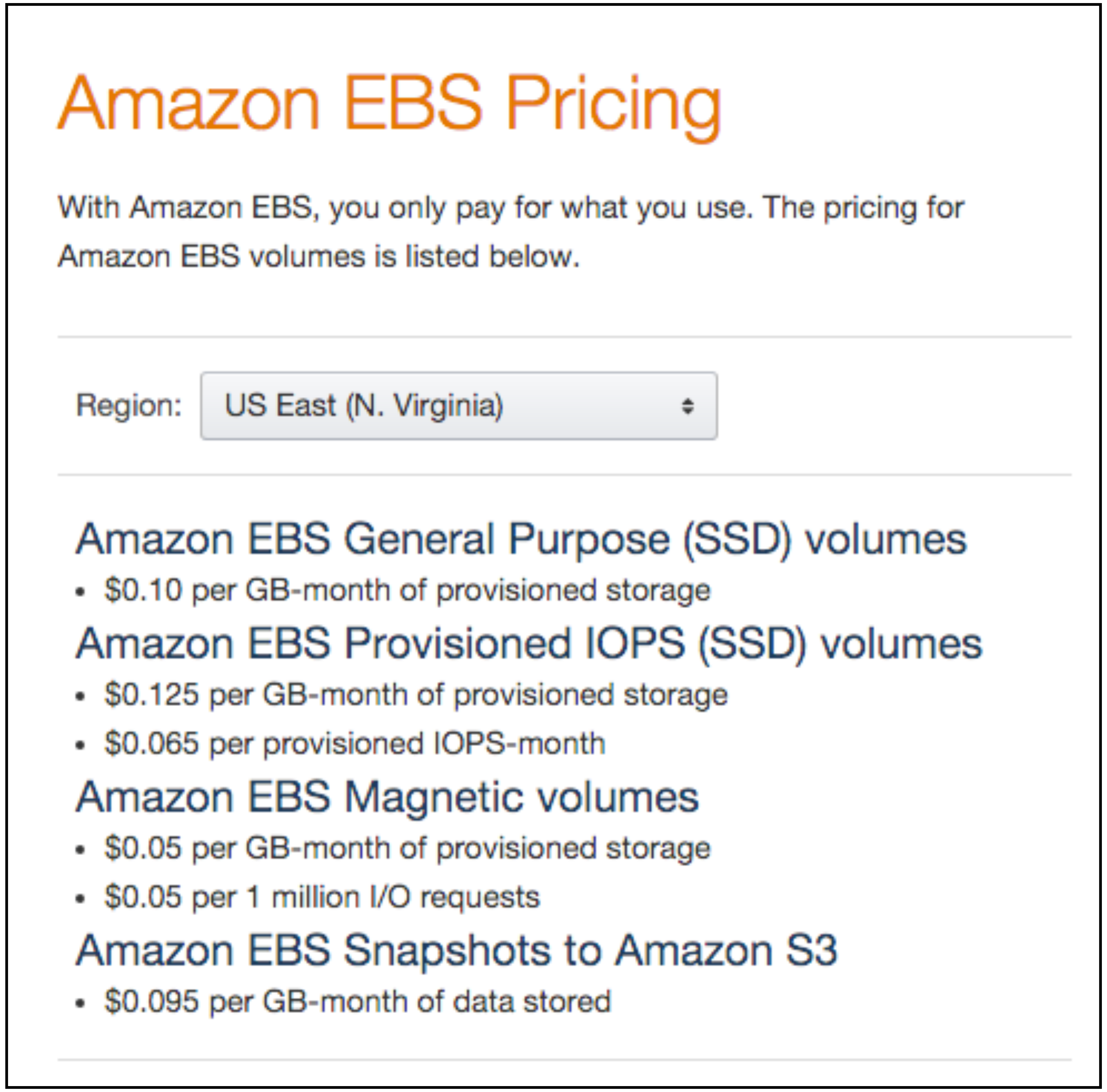
- Boot drives
- Read-only drives to share static databases. (Make 1 TB drive and mount)
- Database drives for MySQL, etc.. (But you should use Amazon’s managed service.)

EBS offers three classes of service.

Characteristic	General Purpose (SSD)	Provisioned IOPS (SSD)	Magnetic
Use cases	- System boot volumes - Virtual desktops - Small to medium sized databases - Development and test environments	- Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume - Large database workloads, such as: ○ MongoDB ○ Microsoft SQL Server ○ MySQL ○ PostgreSQL ○ Oracle	- Cold workloads where data is infrequently accessed - Scenarios where the lowest storage cost is important
Volume size	1 GiB – 16 TiB	4 GiB – 16 TiB	1 GiB – 1 TiB
Maximum throughput	160 MiB/s	320 MiB/s	40–90 MiB/s
IOPS performance	Baseline performance of 3 IOPS/GiB (up to 10,000 IOPS) with the ability to burst to 3,000 IOPS for volumes under 1,000 GiB.	Consistently performs at provisioned level, up to 20,000 IOPS maximum	Averages 100 IOPS, with the ability to burst to hundreds of IOPS
API and CLI volume name	`gp2`	`io1`	`standard`

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

Pricing — you are probably best off with SSD General Purpose.



Amazon EBS Pricing

With Amazon EBS, you only pay for what you use. The pricing for Amazon EBS volumes is listed below.

Region: US East (N. Virginia)

Amazon EBS General Purpose (SSD) volumes

- $0.10 per GB-month of provisioned storage

Amazon EBS Provisioned IOPS (SSD) volumes

- $0.125 per GB-month of provisioned storage
- $0.065 per provisioned IOPS-month

Amazon EBS Magnetic volumes

- $0.05 per GB-month of provisioned storage
- $0.05 per 1 million I/O requests

Amazon EBS Snapshots to Amazon S3

- $0.095 per GB-month of data stored

EBS volumes can be created and used for: extra storage, sharing data

Each EBS volume has:

• Size	e.g. 40GB
• Name	e.g. vol-65202e2d
• Region / AvailabilityZone	e.g. us-east-1 / us-east-1b
• Attributes	e.g. CreateTime, Encrypted, Iops,

Volumes can be mounted:

- read/write on a single instance
- read-only on multiple instances

Create and share an instance:

```
$ aws ec2 create-volume --size 10 --availability-zone us-east-1a
You must specify a region. You can also configure your region by running "aws configure".
$ aws ec2 create-volume --size 10 --region us-east-1 --availability-zone us-east-1b
{
    "AvailabilityZone": "us-east-1b",
    "Encrypted": false,
    "VolumeType": "standard",
    "VolumeId": "vol-95cab176",
    "State": "creating",
    "SnapshotId": "",
    "CreateTime": "2015-12-05T18:55:28.052Z",
    "Size": 10
}
$
```

Attach the EBS volume to your VM
(Be sure EBS is in same region & availability zone)

First get a volume...

```
$ aws_zone=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)
$ aws_instance=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)
$ aws_region=$(curl -s http://169.254.169.254/latest/dynamic/instance-identity/document|grep region|awk -F\" '{print $4}')
$ aws ec2 create-volume --size 10 --region $aws_region --availability-zone $aws_zone
{
    "AvailabilityZone": "us-east-1b",
    "Encrypted": false,
    "VolumeType": "standard",
    "VolumeId": "vol-46cdb6a5",
    "State": "creating",
    "SnapshotId": "",
    "CreateTime": "2015-12-05T19:01:38.548Z",
    "Size": 10
}
$ aws ec2 attach-volume --volume-id=vol-46cdb6a5 --instance-id=$aws_instance \
  --device=/dev/sdb --region=$aws_region
{
    "AttachTime": "2015-12-05T19:02:11.541Z",
    "InstanceId": "i-5c306beb",
    "VolumeId": "vol-46cdb6a5",
    "State": "attaching",
    "Device": "/dev/sdb"
}
$
```

Now we need to make a file system...

Create a file system on the volume

```
$ sudo mkfs -t ext4 /dev/sdb
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 2621440 4k blocks and 655360 inodes
Filesystem UUID: 681c57f0-1461-4dae-b956-032656ba82a9
Superblock backups stored on blocks:
     32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

$ sudo mount /dev/sdb /mnt/extra/
[ip-172-30-1-33 ~ 19:04:44]$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/xvda1      41151788 6506728  34544812  16% /
devtmpfs          500712      60    500652   1% /dev
tmpfs             509724       0    509724   0% /dev/shm
/dev/xvdb       10190136   23028   9626436   1% /mnt/extra

$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  40G  0 disk
└─xvda1 202:1    0  40G  0 part /
xvdb    202:16   0  10G  0 disk /mnt/extra
$
```

EBS Snapshots: Sharing between multiple systems (and users)

EBS volumes: only mounted read/write on one instance at a time.

- Most file systems don’t support multiple writers from different systems.
- Weird consistency issues in a networked environment.

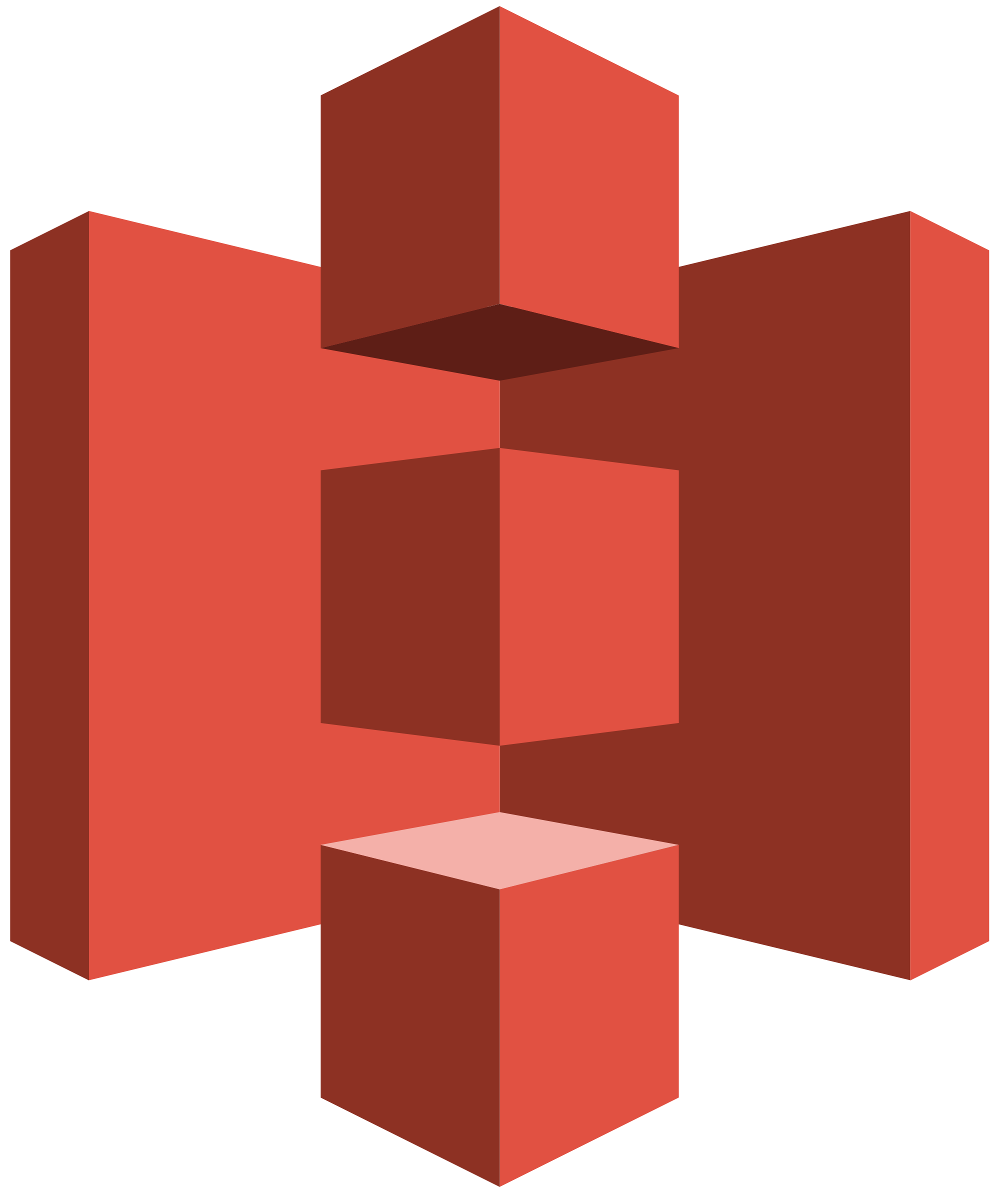
Snapshots allow:

- A single read-only volume to be mounted by many users.
- Publishing an EBS volume to others.
- Restore to a different volume.

Information on snapshots:

- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html
- http://angus.readthedocs.org/en/2014/amazon/using-ebs-snapshot.html

Amazon S3



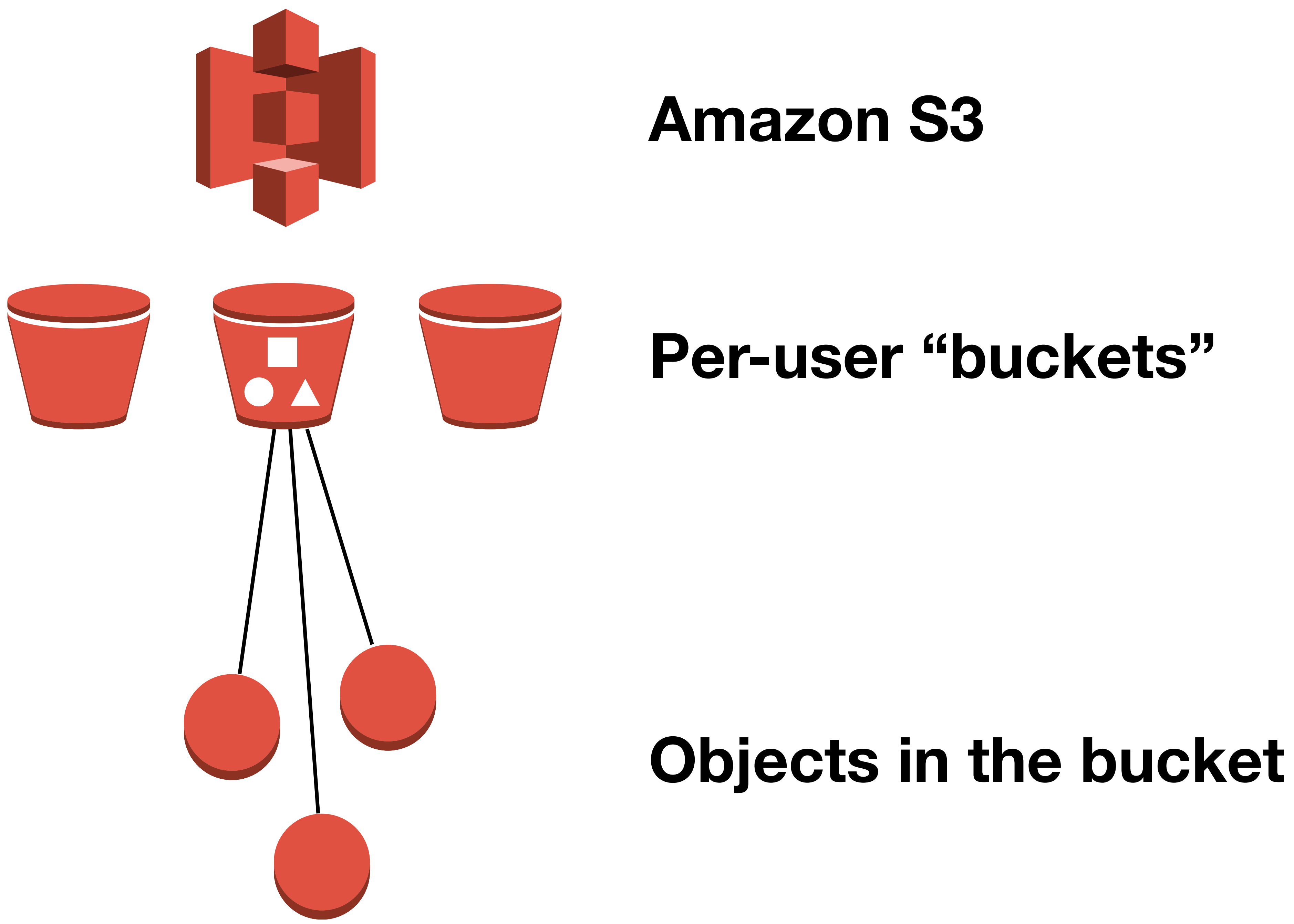
S3 is an object-based storage system

Every S3 bucket has:

- Name
- Owner
- Access permissions

Every S3 object has:

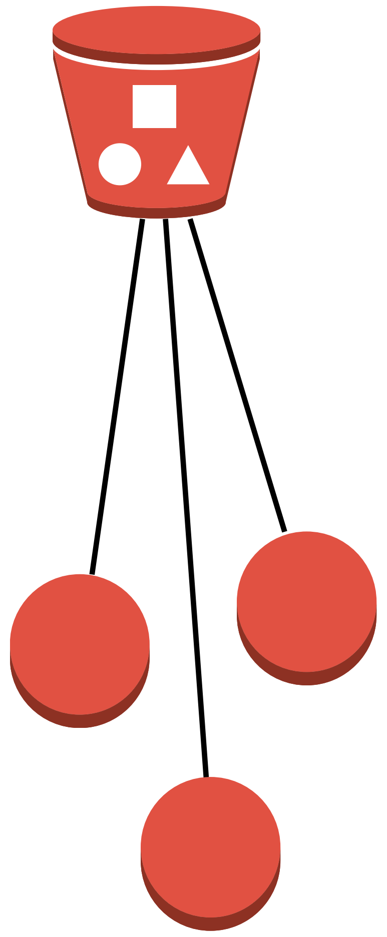
- Size
- URL
- Access permissions
  - —*e.g. world readable*



Accessing S3 data

Uses of S3:

- Storing logs
- Distributing data



Advantages of S3:

- *permanence; S3 outlasts your EC2/EMR cluster*
- *Pay only for what you need, rather than for virtual drives capacity.*

Disadvantage of S3

- *No data locality. S3 data always moves over the network.*
- *Just like EBS!*
- *Not a big issue with 10g instances.*

Remember: S3 is not a file system, it's an object storage system.

- https://wiki.apache.org/hadoop/AmazonS3

Amazon Security

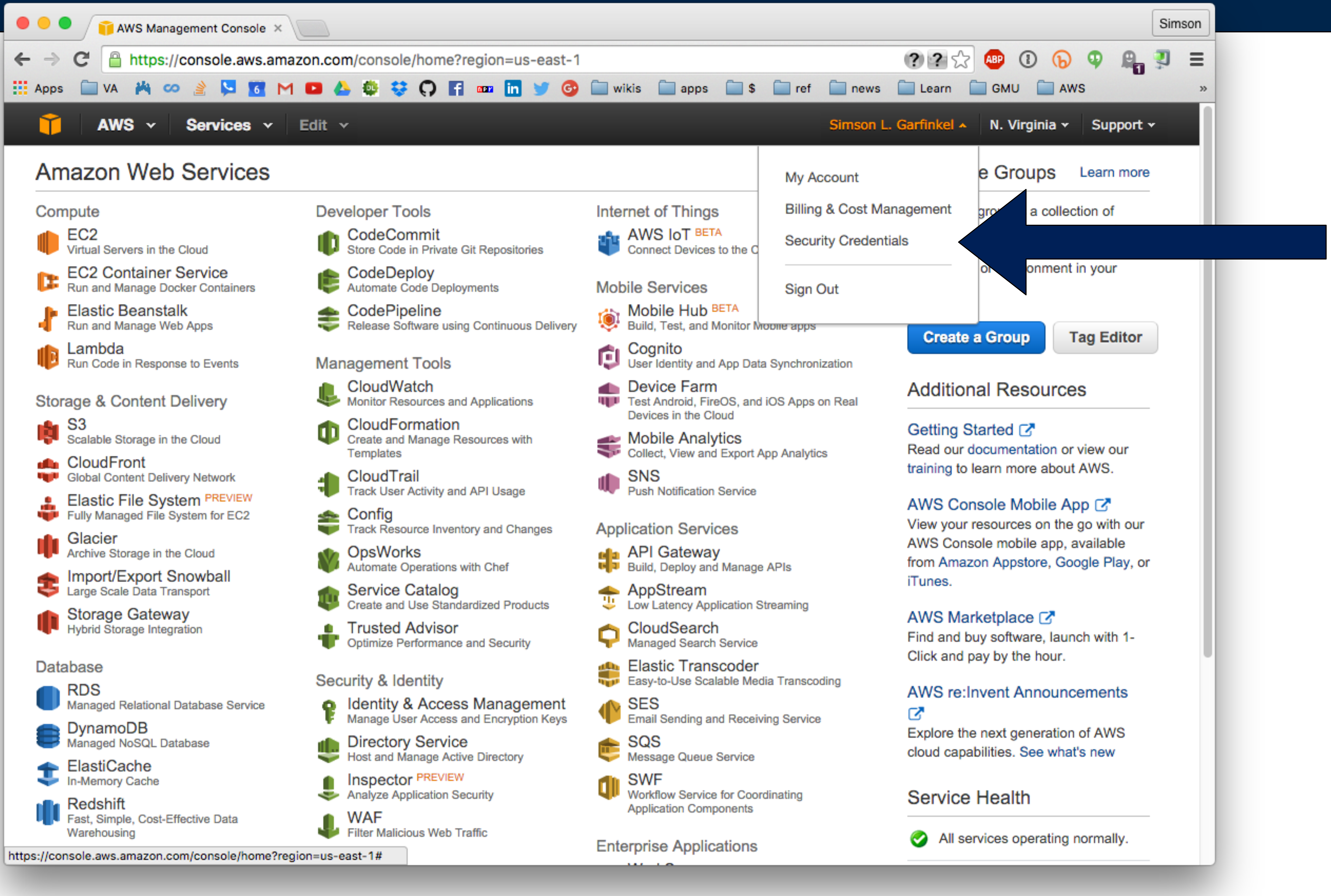
Protecting your account

Complex password.

Two-factor authentication

Account vs. Instance Credentials

AWS_KEY vs. AWS_SECRET_KEY



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 BETA – Connect Devices to the C

Mobile Services

- Mobile Hub BETA – 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 – Message Queue Service
- SWF – Workflow Service for Coordinating Application Components

Enterprise Applications

e Groups

Learn more

gro a collection of

o onment in your

Additional Resources

Getting Started
Read our documentation or view our training to learn more about AWS.

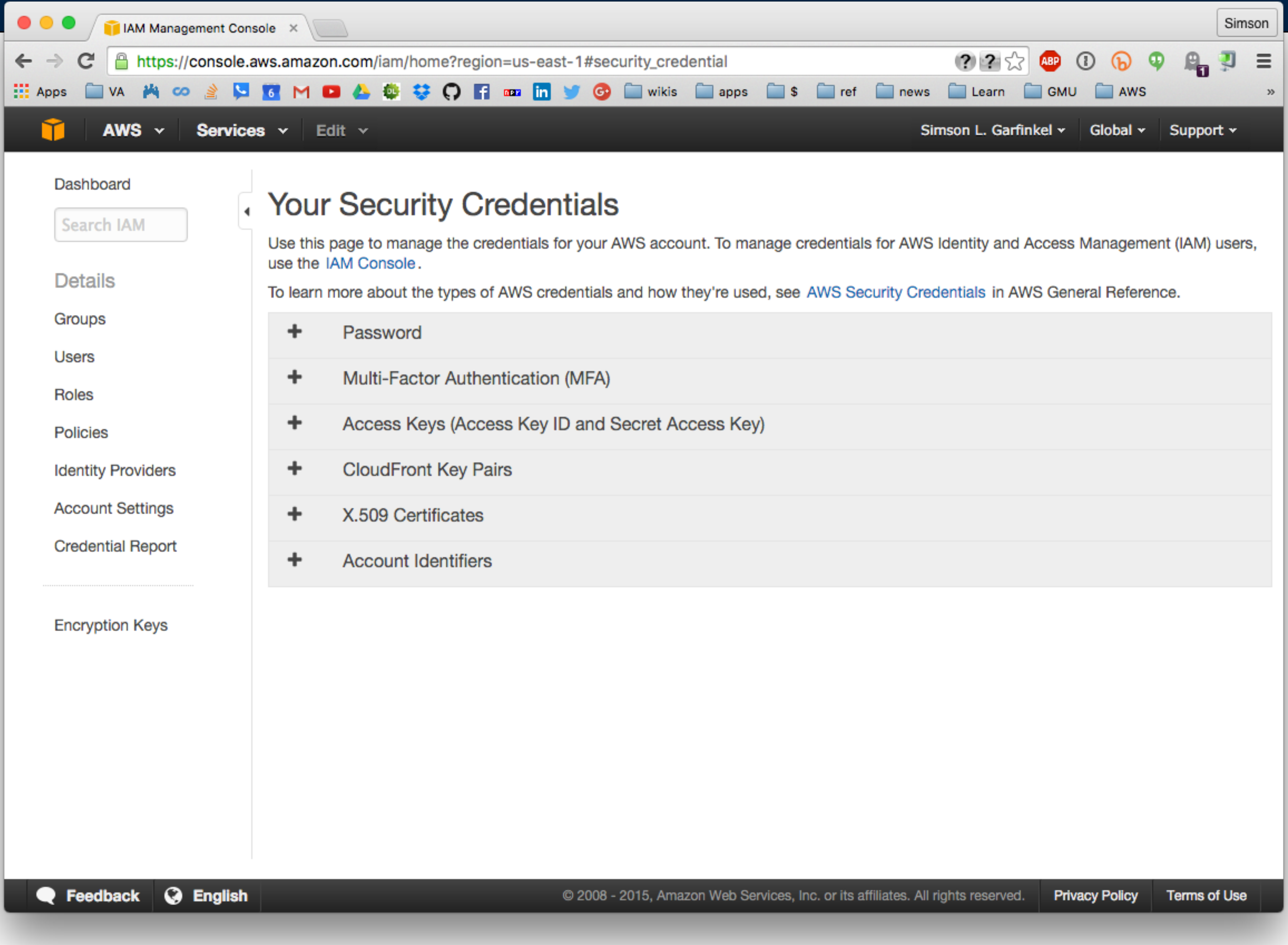
AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

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.



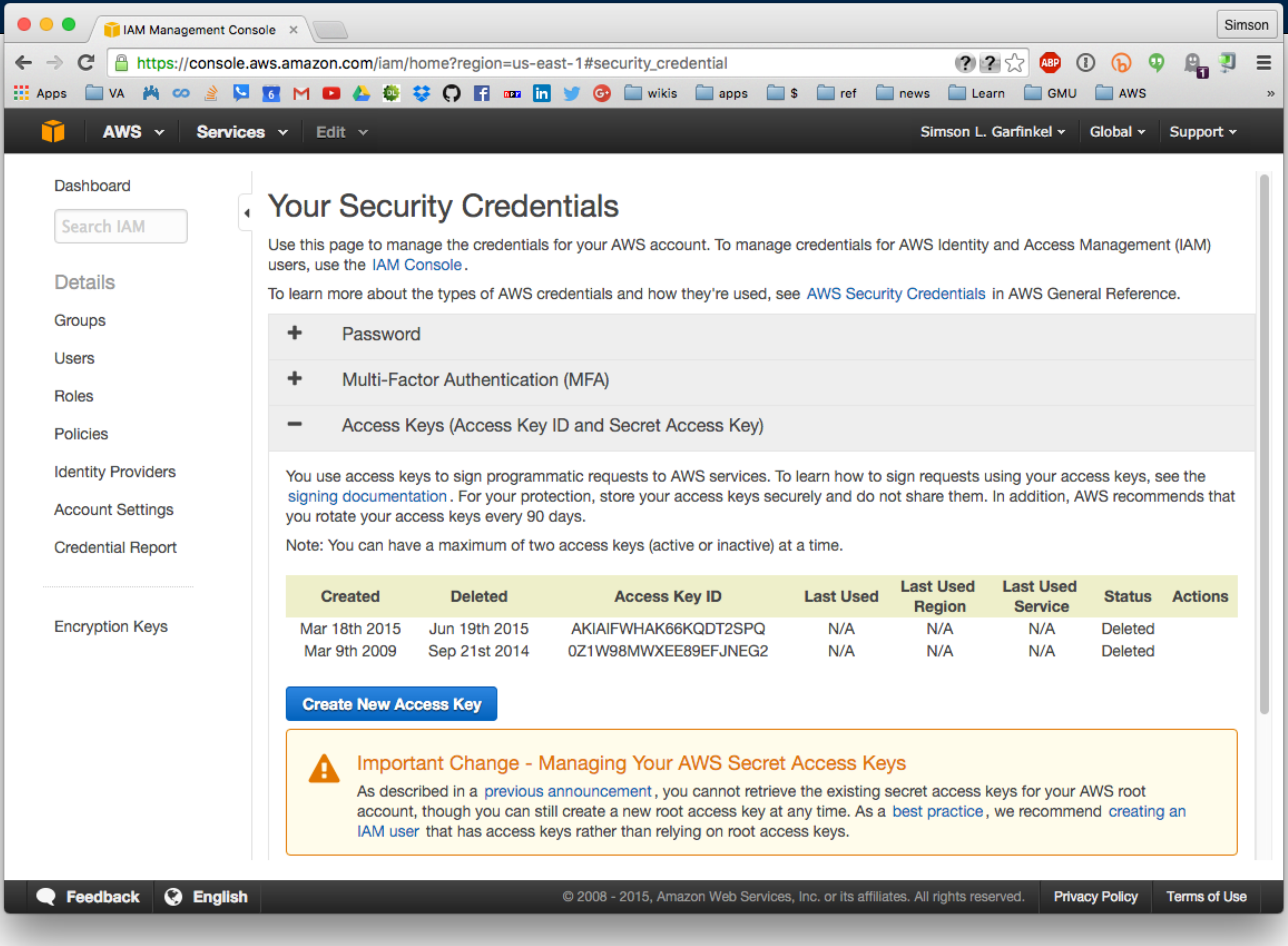
Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

- ✚ Password
- ✚ Multi-Factor Authentication (MFA)
- ✚ Access Keys (Access Key ID and Secret Access Key)
- ✚ CloudFront Key Pairs
- ✚ X.509 Certificates
- ✚ Account Identifiers

Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

+ Password

+ Multi-Factor Authentication (MFA)

− Access Keys (Access Key ID and Secret Access Key)

You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the signing documentation. For your protection, store your access keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90 days.

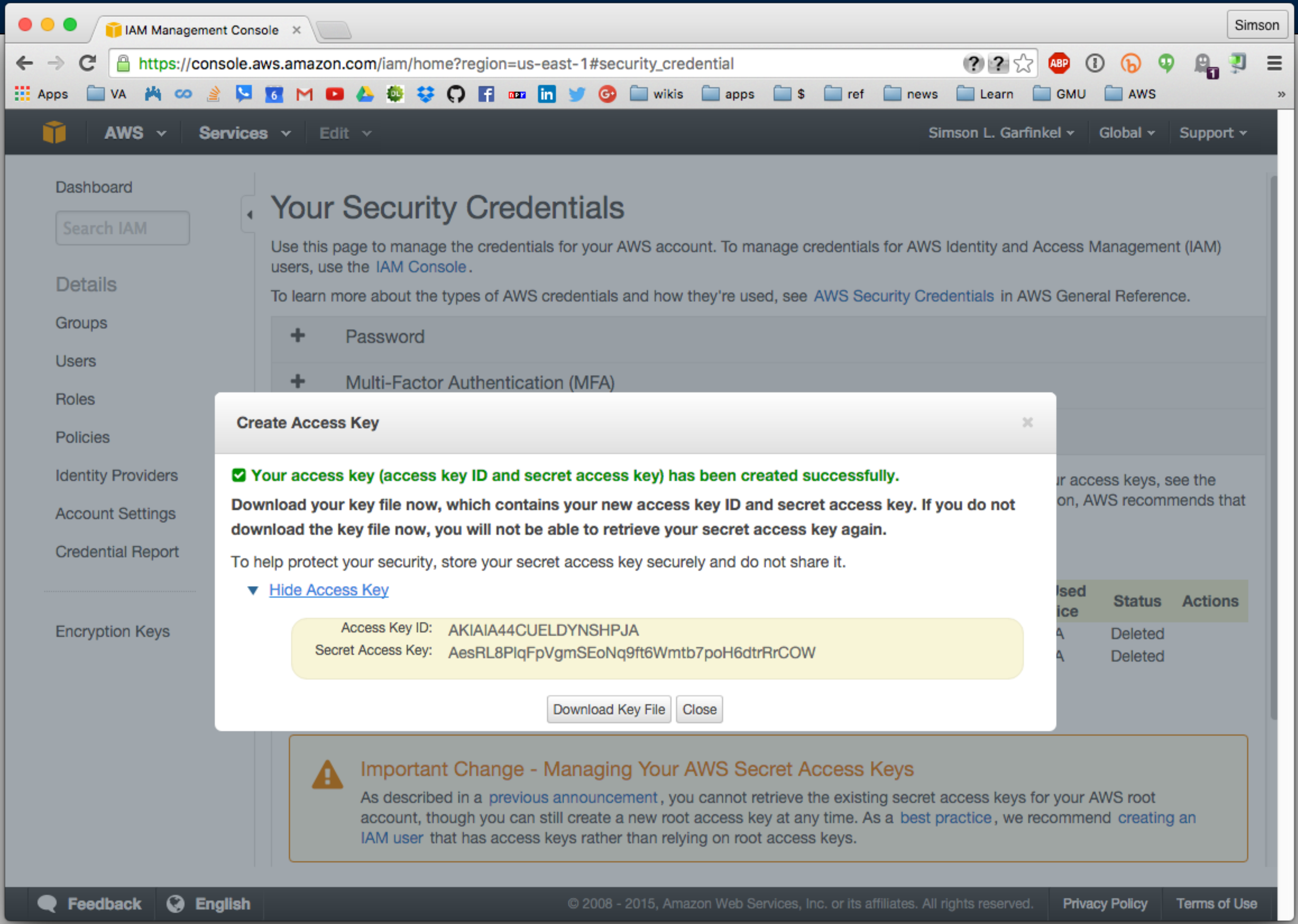
Note: You can have a maximum of two access keys (active or inactive) at a time.

Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Mar 18th 2015	Jun 19th 2015	AKIAIFWHAK66KQDT2SPQ	N/A	N/A	N/A	Deleted	
Mar 9th 2009	Sep 21st 2014	0Z1W98MWXEE89EFJNEG2	N/A	N/A	N/A	Deleted	

Create New Access Key

Important Change - Managing Your AWS Secret Access Keys

As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

+ Password

+ Multi-Factor Authentication (MFA)

ur access keys, see the
on, AWS recommends that

sed ice	Status	Actions
A	Deleted	
A	Deleted	

Create Access Key

☑ Your access key (access key ID and secret access key) has been created successfully.

Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.

To help protect your security, store your secret access key securely and do not share it.

▼ Hide Access Key

Access Key ID: AKIAIA44CUELDYNSHPJA
Secret Access Key: AesRL8PlqFpVgmSEoNq9ft6Wmtb7poH6dtrRrCOW

Download Key File | Close

Important Change - Managing Your AWS Secret Access Keys

As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

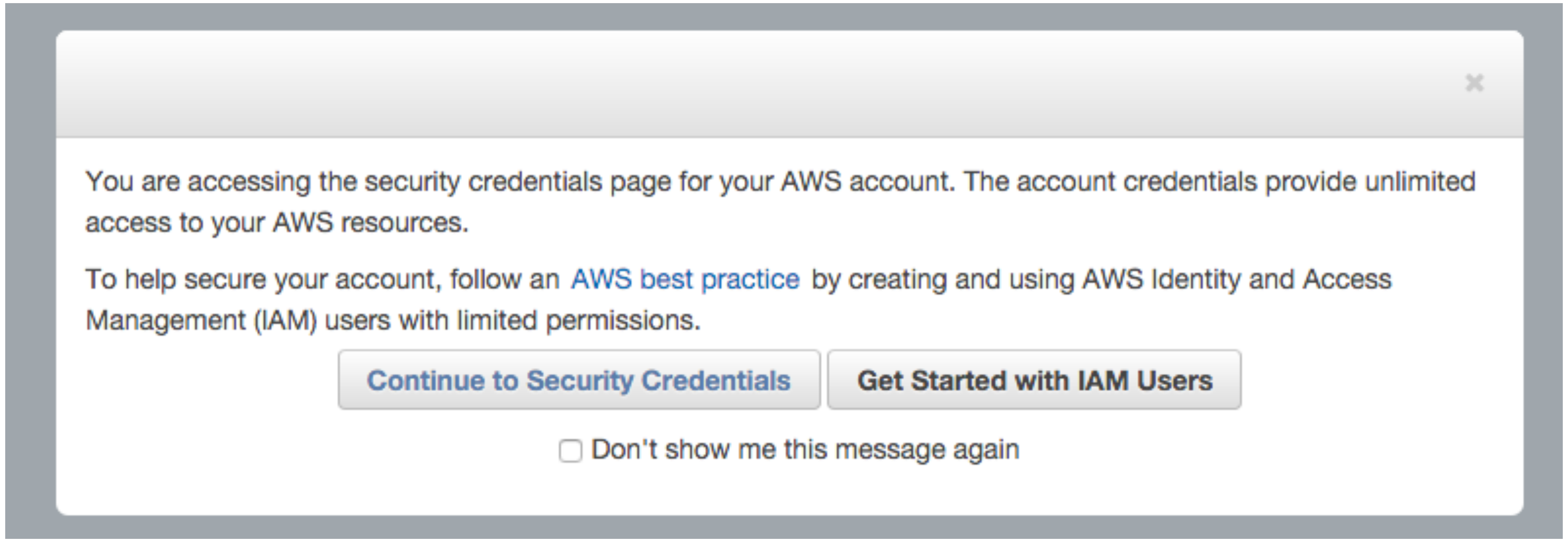
Best practices

Access keys:

- Don’t store in code. Store in an AWS credential file, environment variables, ~/.boto file, etc.

IAM users:

- Have their own username & password
- Can have authentication “burned in” to a EC2 instance
    - http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html



You are accessing the security credentials page for your AWS account. The account credentials provide unlimited access to your AWS resources.

To help secure your account, follow an AWS best practice by creating and using AWS Identity and Access Management (IAM) users with limited permissions.

×
Continue to Security Credentials
Get Started with IAM Users
□ Don't show me this message again

Security Credentials

Don’t use your AWS account access keys; create an IAM account.

Each IAM account has its own:

- Username, password, etc.
- Groups and Policies (what it can do)
- Access Keys (for API control.)
- It’s own multi-factor authentication.

Why use IAM accounts?

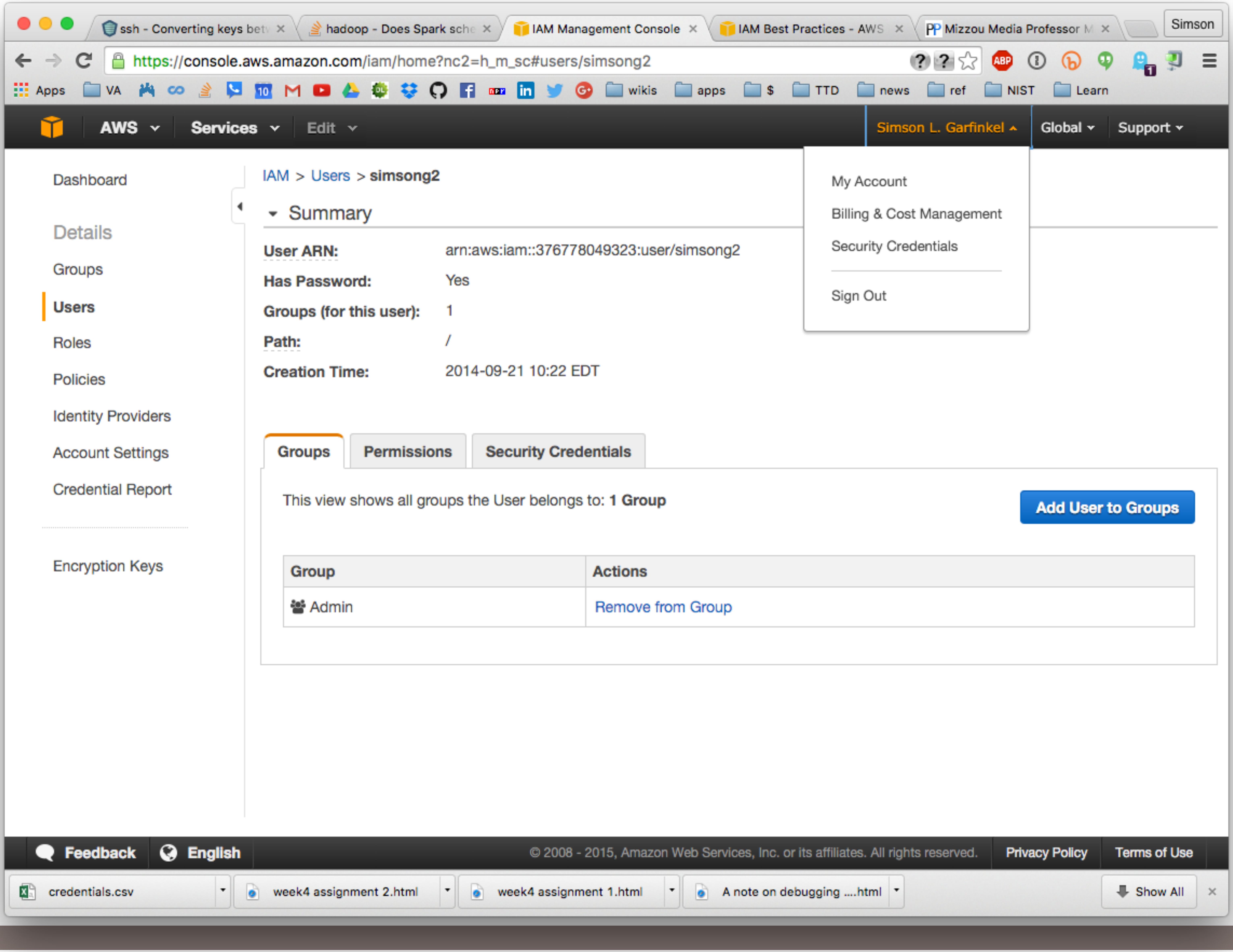
- Prevents a single user from wiping your entire cluster, data, etc.
- Allows you to create users that can only do a few things, without changing configurations:
  - —*View data*
  - —*Update data*

When you use IAM, you have a different name and login

My accounts:

- Account name: simsong
- Account sign in: https://simsong.signin.aws.amazon.com/console

By default, new users are in the “Admin” group



https://console.aws.amazon.com/iam/home?nc2=h_m_sc#users/simsong2

IAM > Users > simsong2

Summary

User ARN: arn:aws:iam::376778049323:user/simsong2

Has Password: Yes

Groups (for this user): 1

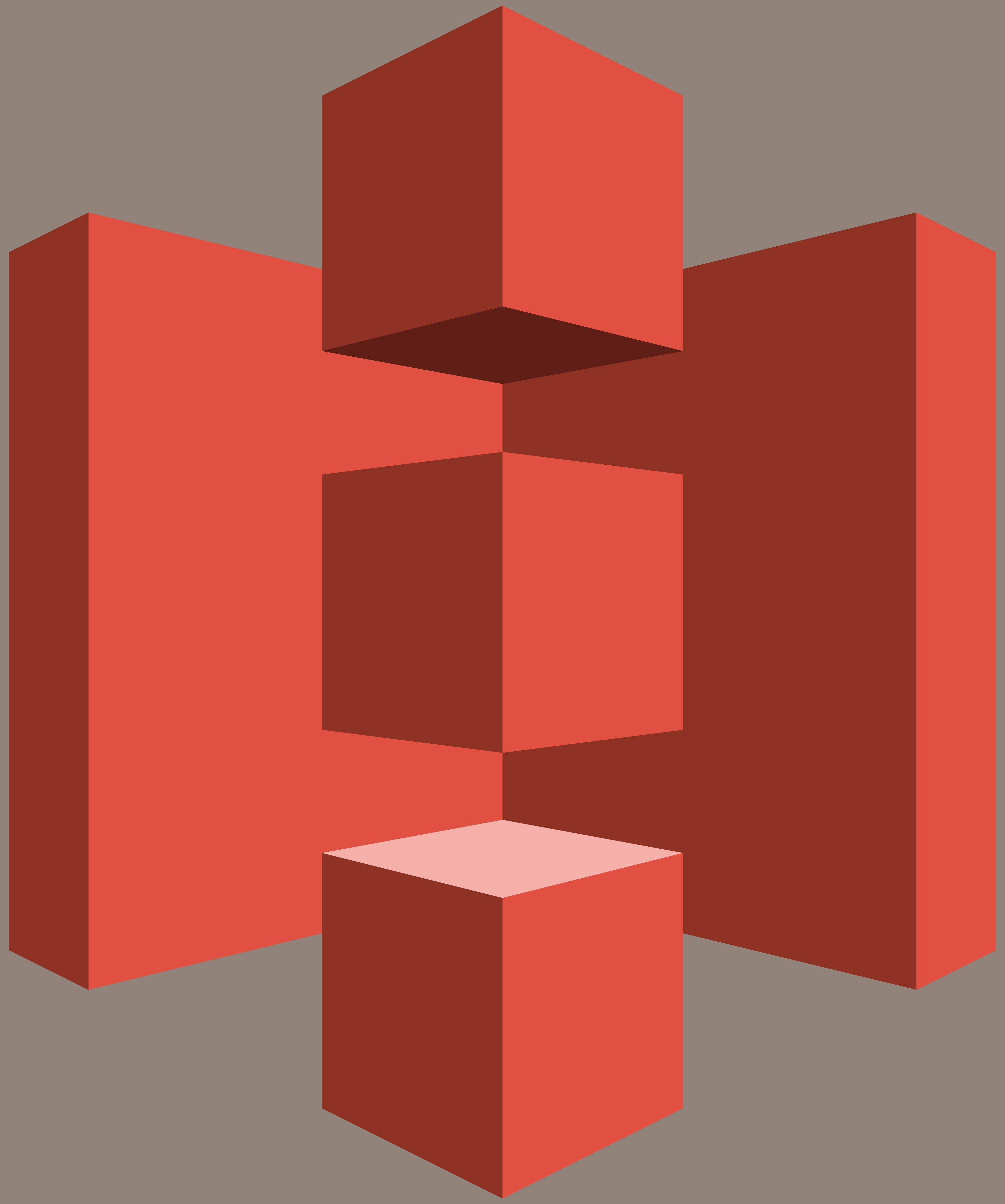
Path: /

Creation Time: 2014-09-21 10:22 EDT

This view shows all groups the User belongs to: **1 Group**

Group	Actions
Admin	Remove from Group

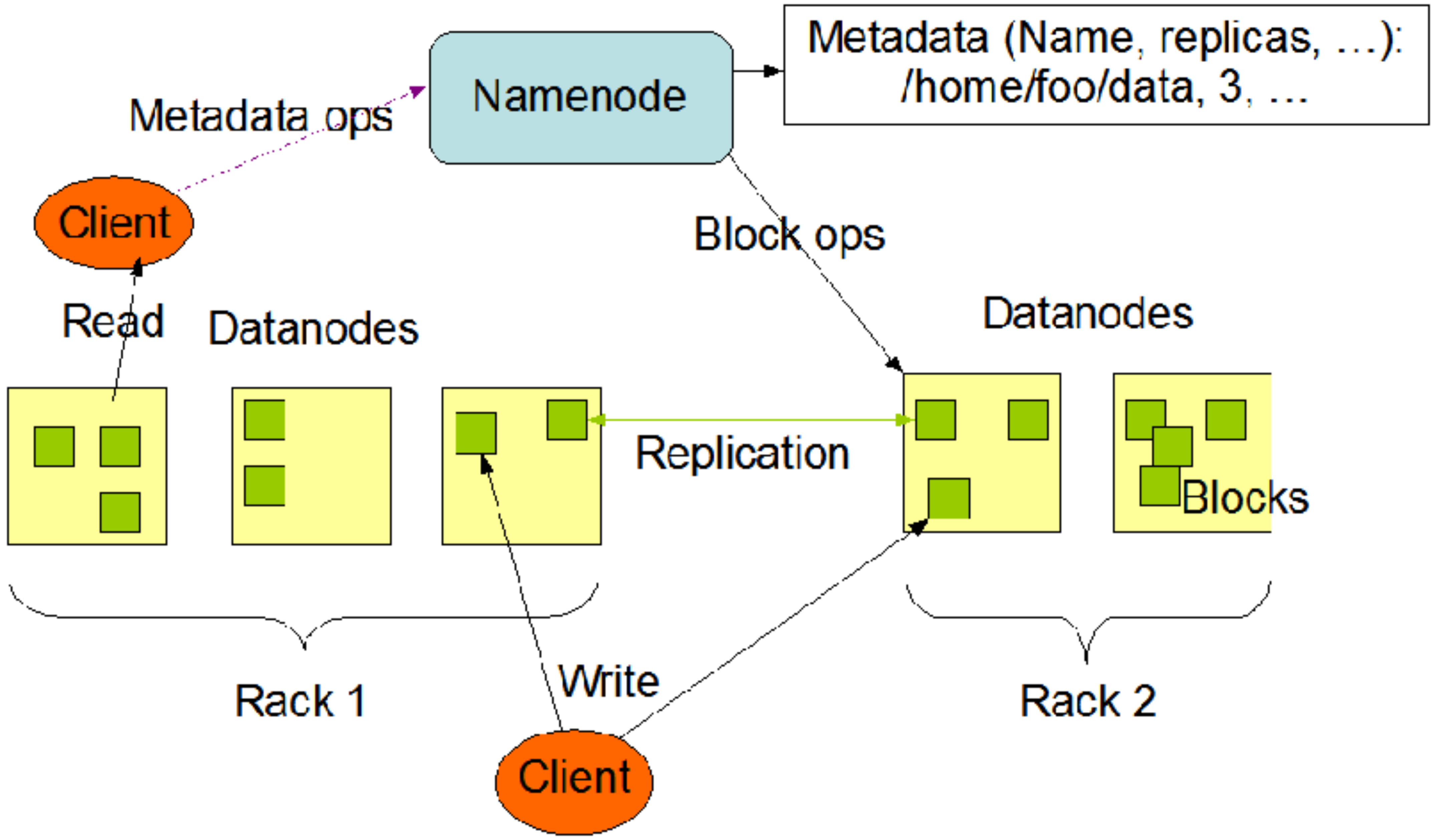
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.



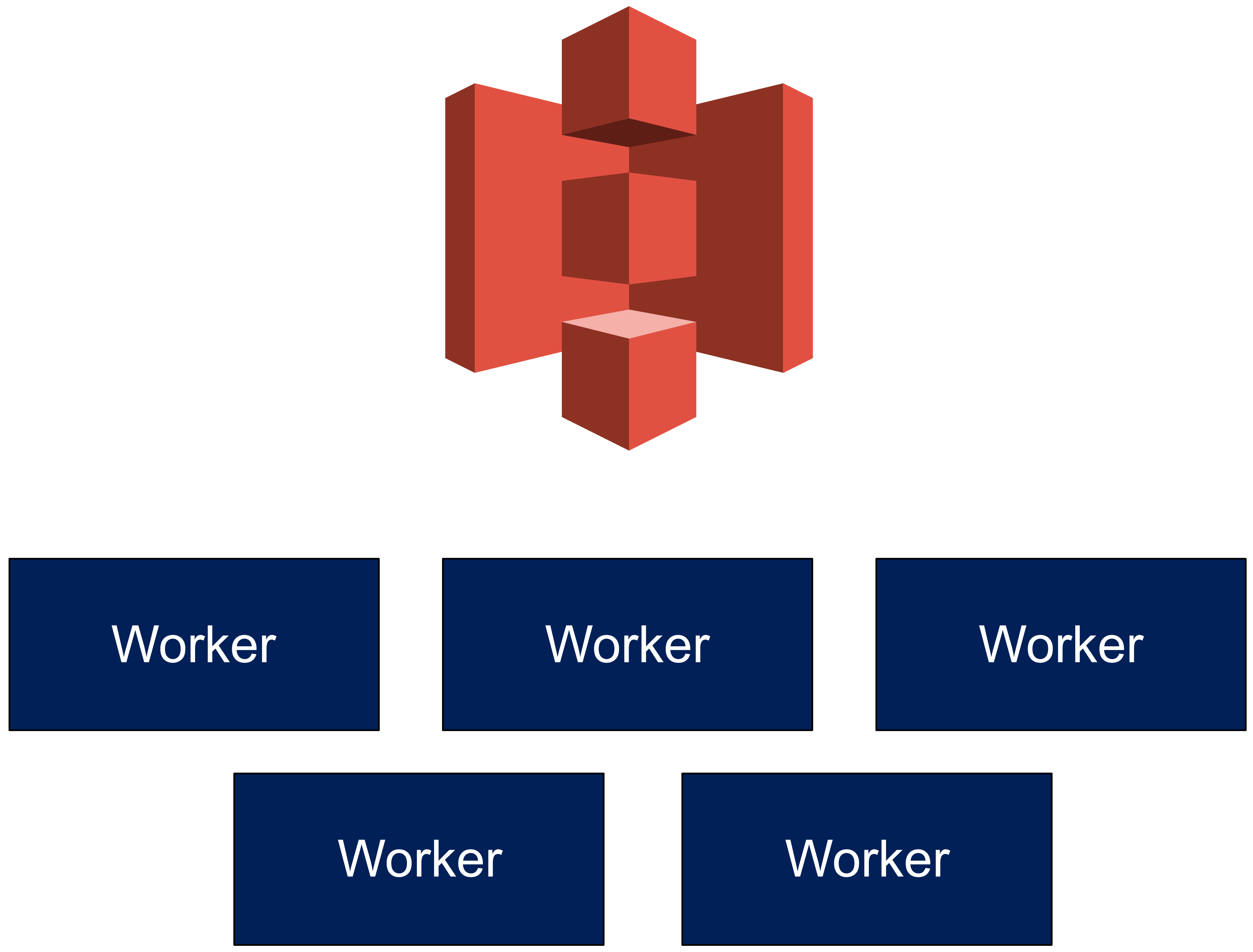
Amazon S3 — Deep Dive

S3 vs. HDFS

HDFS:



Amazon S3:



Worker

Worker

Worker

Worker

Worker

S3 arranges objects (files) into “buckets.” Each user has one or more buckets

Each bucket:

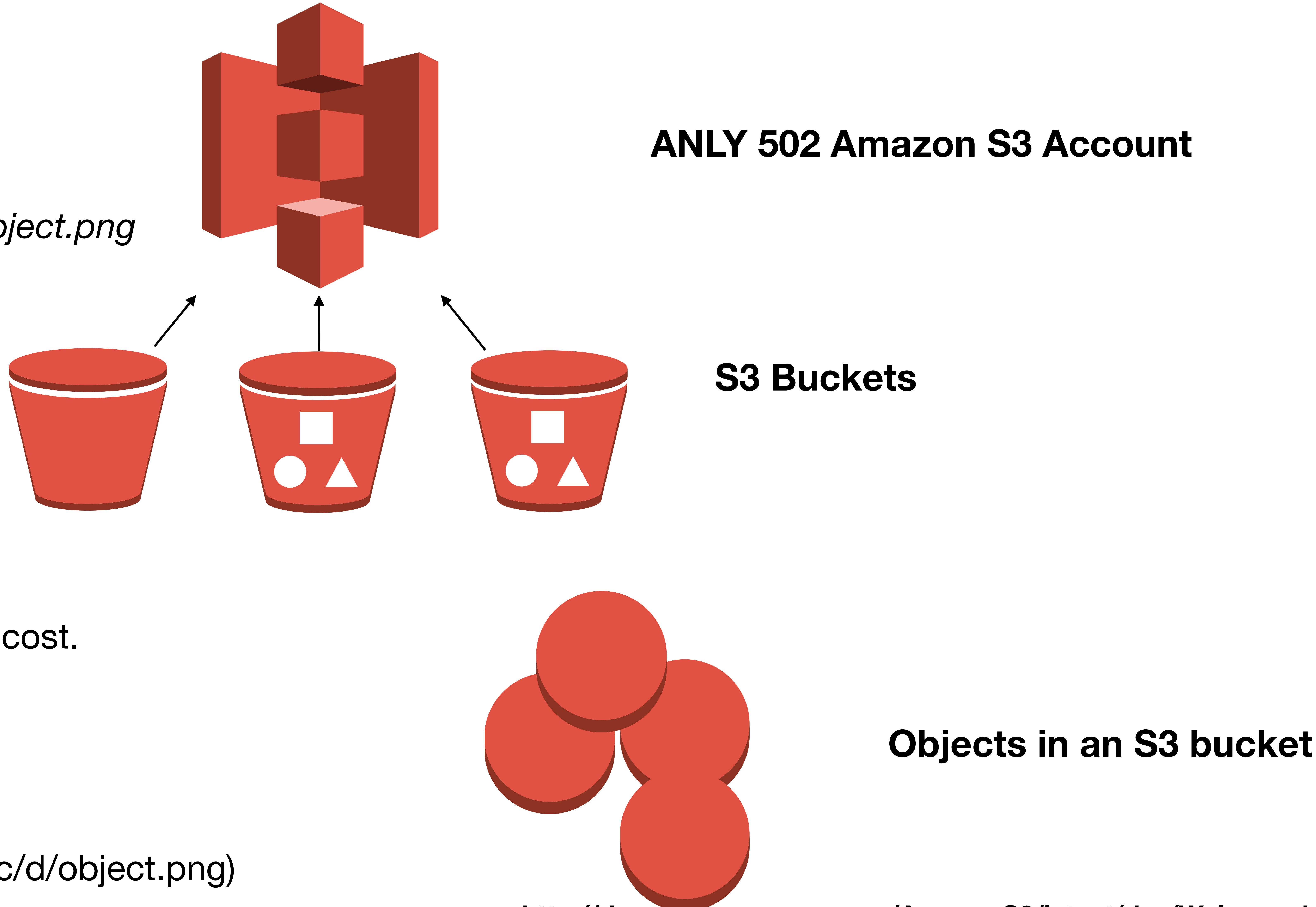
- Has a unique name
- Has a unique URL
  - https://anly502.s3.amazonaws.com/object.png

Buckets can also:

- Set to a specific region
- Enable versioning
- Serve static HTML pages.
- Multiple consistency models.
- Reduced Redundancy Storage for lower cost.

Objects:

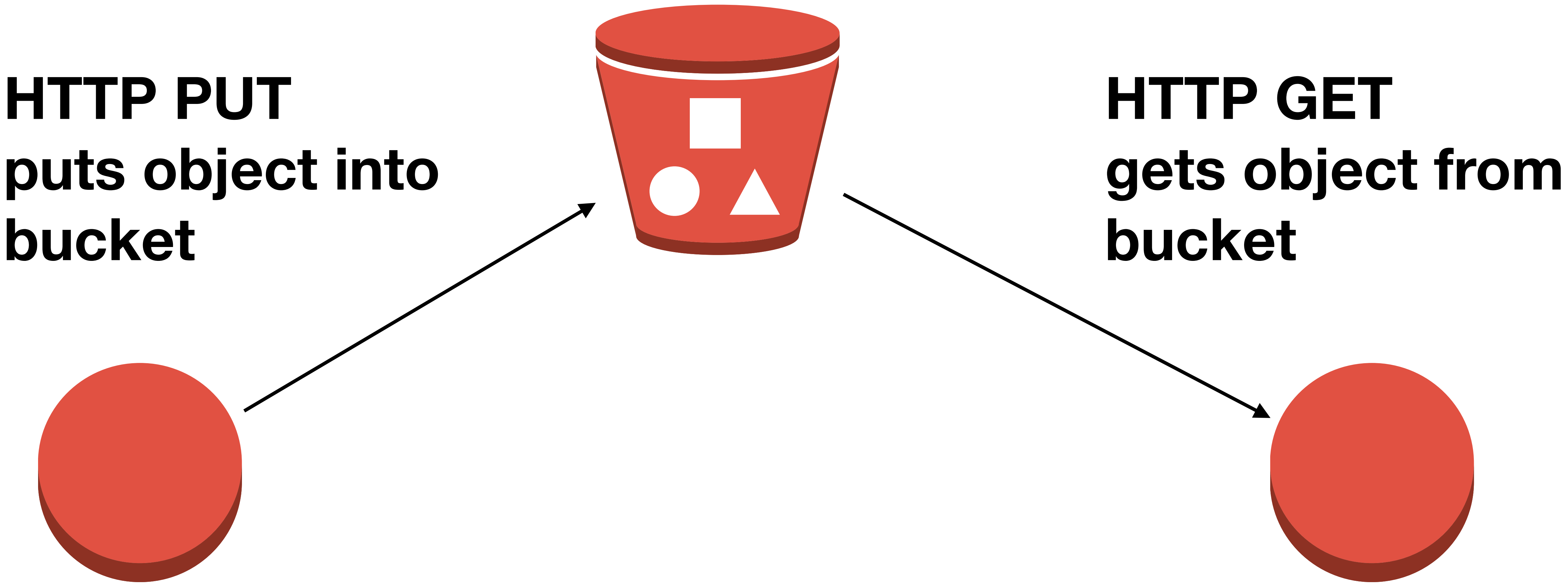
- In buckets
- Identified by “keys” (e.g object.png, a/b/c/d/object.png)



http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html

S3 access protocol: REST

REST is built on top of HTTP.



HTTP PUT
puts object into bucket

HTTP GET
gets object from bucket

Many ways to access data on Amazon S3

HTTP / REST — Representational State Transfer

- Uses HTTP methods (with a bit of JSON)
- HTTP GET — Reads a resource without causing any side effects
- HTTP DELETE — Deletes a resources
- HTTP PUT (or POST) — Creates a new resources
- HTTP POST (or PUT) — Modify a resource’s value

HTTP Hosting

- Different from REST
- Must be explicitly enabled

HTTP / SOAP — Simple Object Access Protocol

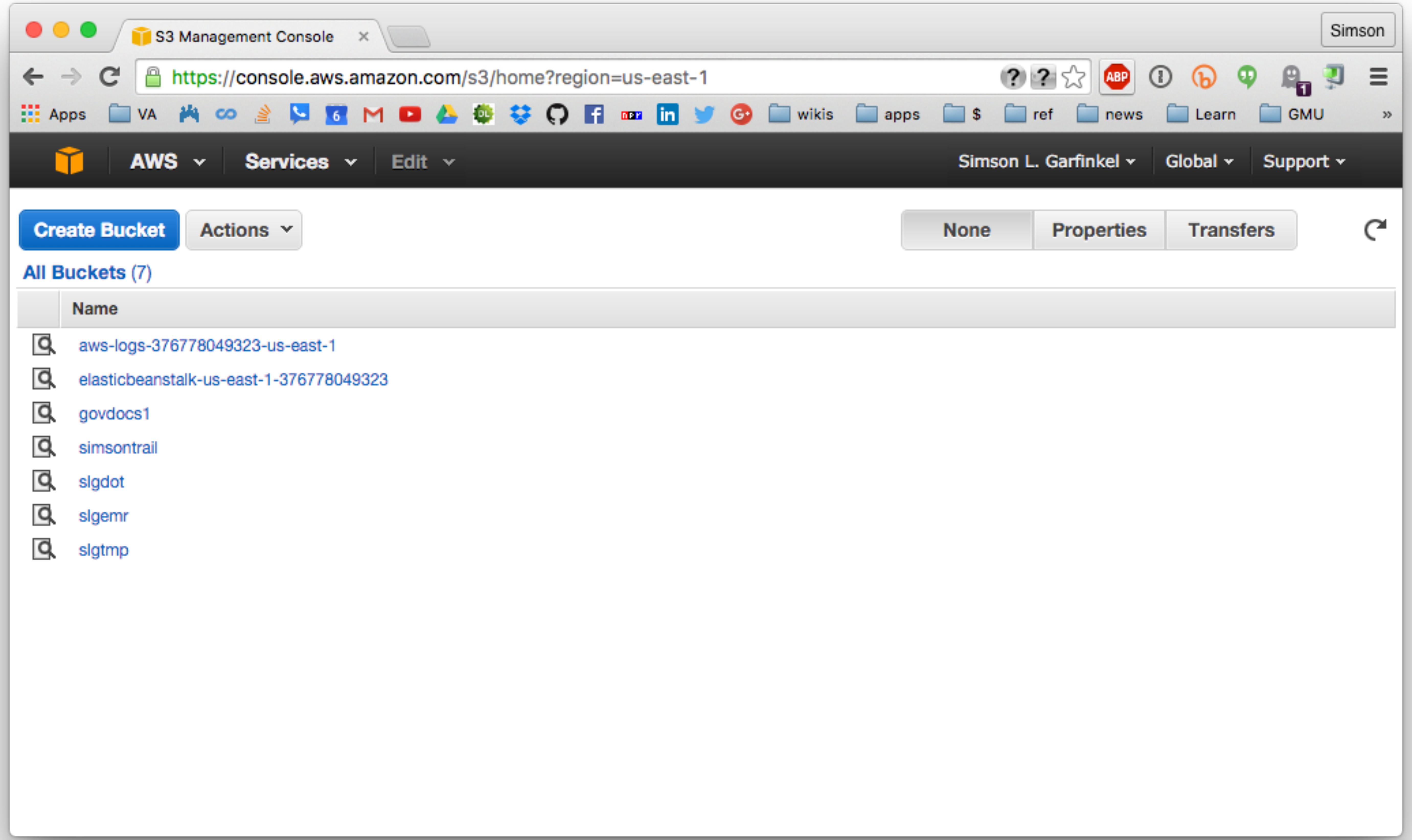
- Structure XML-based protocol
- Heavy weight; increasingly not used.

BitTorrent

- S3 can host a “tracker” and “seeds”
- Limited to objects 5GB in size

Buckets are controlled from web-API or CLI

View all buckets



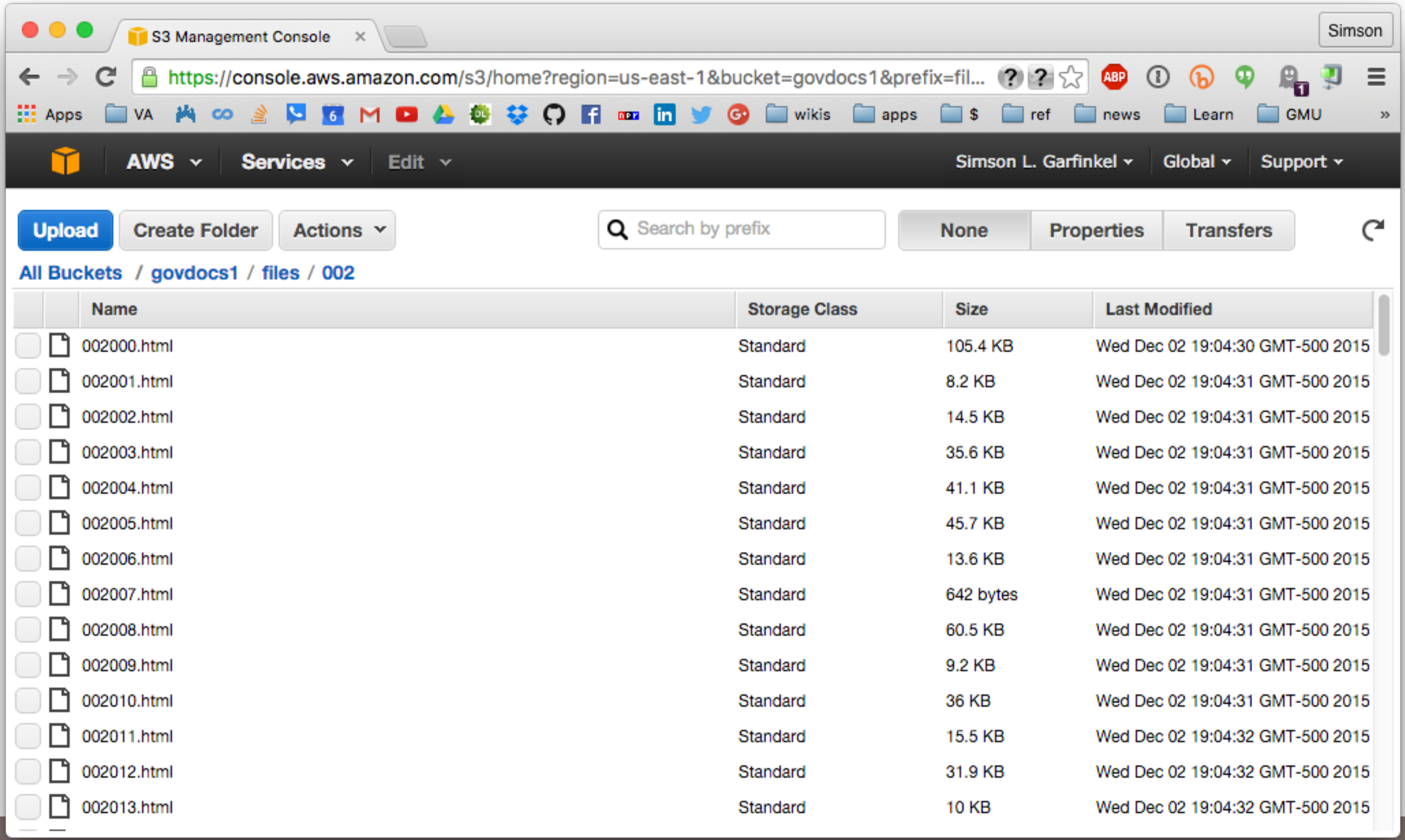
All Buckets (7)

Name
aws-logs-376778049323-us-east-1
elasticbeanstalk-us-east-1-376778049323
govdocs1
simsontrail
slgdot
slgemr
slgtmp

Drill down to a specific file in a bucket

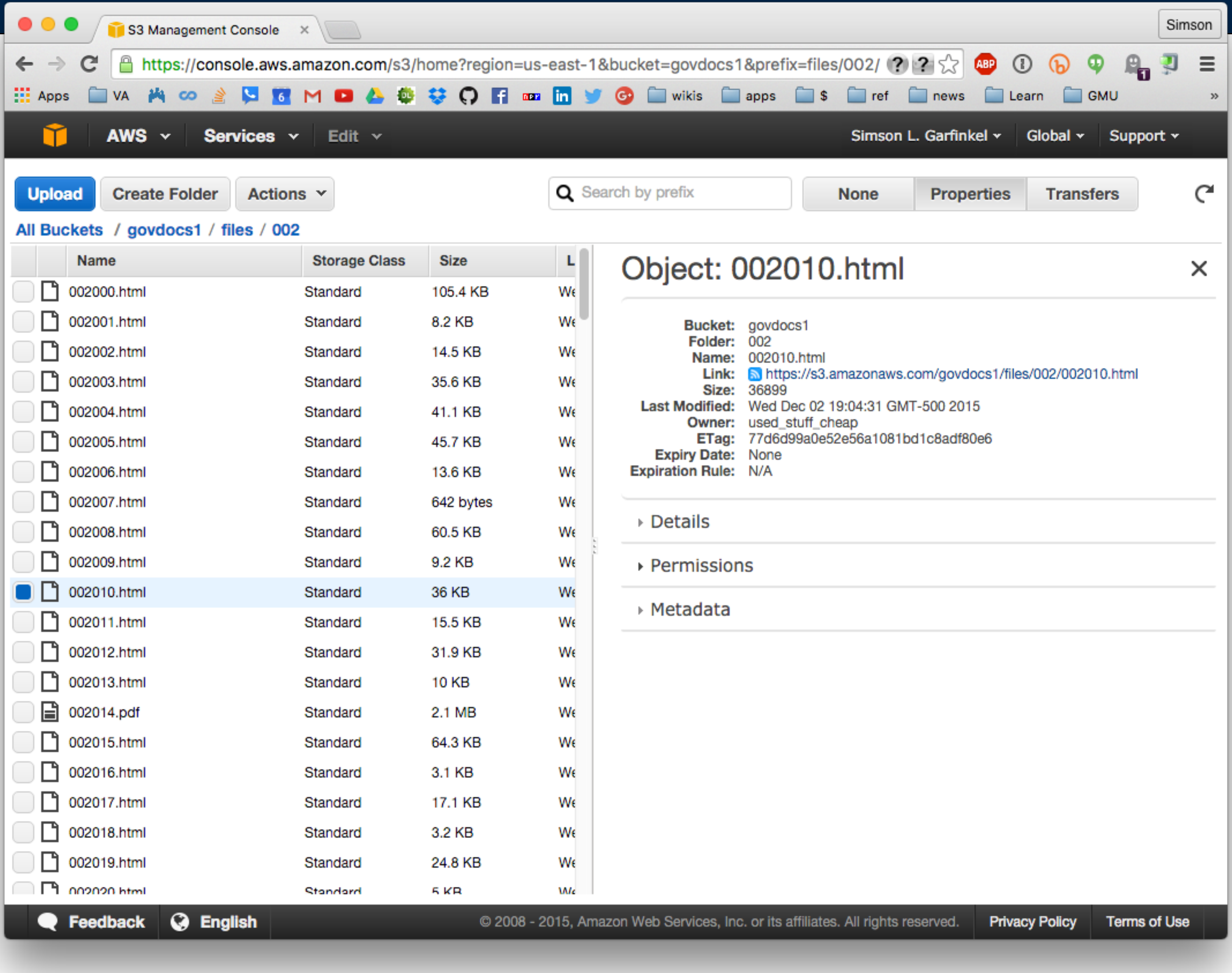
Folders don’t actually exist.

- “For the sake of organizational simplicity, the Amazon S3 console supports the folder concept as a means of grouping objects. Amazon S3 does this by using key name prefixes for objects.”
    - http://docs.aws.amazon.com/AmazonS3/latest/UG/FolderOperations.html
- Remember: govdocs1/files/002 is not a directory — it’s a prefix



Name	Storage Class	Size	Last Modified
002000.html	Standard	105.4 KB	Wed Dec 02 19:04:30 GMT-500 2015
002001.html	Standard	8.2 KB	Wed Dec 02 19:04:31 GMT-500 2015
002002.html	Standard	14.5 KB	Wed Dec 02 19:04:31 GMT-500 2015
002003.html	Standard	35.6 KB	Wed Dec 02 19:04:31 GMT-500 2015
002004.html	Standard	41.1 KB	Wed Dec 02 19:04:31 GMT-500 2015
002005.html	Standard	45.7 KB	Wed Dec 02 19:04:31 GMT-500 2015
002006.html	Standard	13.6 KB	Wed Dec 02 19:04:31 GMT-500 2015
002007.html	Standard	642 bytes	Wed Dec 02 19:04:31 GMT-500 2015
002008.html	Standard	60.5 KB	Wed Dec 02 19:04:31 GMT-500 2015
002009.html	Standard	9.2 KB	Wed Dec 02 19:04:31 GMT-500 2015
002010.html	Standard	36 KB	Wed Dec 02 19:04:31 GMT-500 2015
002011.html	Standard	15.5 KB	Wed Dec 02 19:04:32 GMT-500 2015
002012.html	Standard	31.9 KB	Wed Dec 02 19:04:32 GMT-500 2015
002013.html	Standard	10 KB	Wed Dec 02 19:04:32 GMT-500 2015

Each object has properties

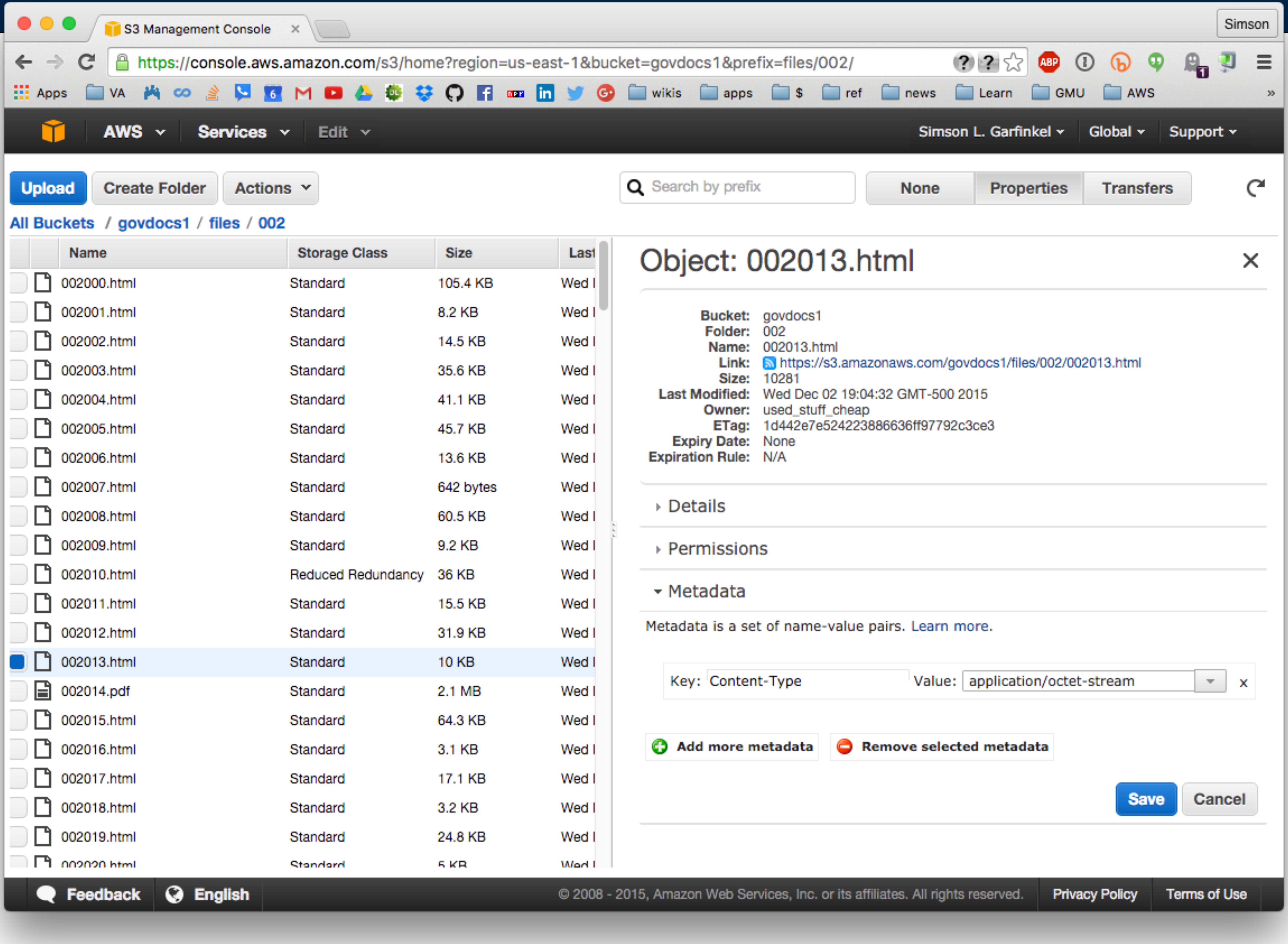


Name	Storage Class	Size	L
002000.html	Standard	105.4 KB	We
002001.html	Standard	8.2 KB	We
002002.html	Standard	14.5 KB	We
002003.html	Standard	35.6 KB	We
002004.html	Standard	41.1 KB	We
002005.html	Standard	45.7 KB	We
002006.html	Standard	13.6 KB	We
002007.html	Standard	642 bytes	We
002008.html	Standard	60.5 KB	We
002009.html	Standard	9.2 KB	We
002010.html	Standard	36 KB	We
002011.html	Standard	15.5 KB	We
002012.html	Standard	31.9 KB	We
002013.html	Standard	10 KB	We
002014.pdf	Standard	2.1 MB	We
002015.html	Standard	64.3 KB	We
002016.html	Standard	3.1 KB	We
002017.html	Standard	17.1 KB	We
002018.html	Standard	3.2 KB	We
002019.html	Standard	24.8 KB	We
002020.html	Standard	5 KB	We

Object: 002010.html

Bucket: govdocs1
Folder: 002
Name: 002010.html
Link: https://s3.amazonaws.com/govdocs1/files/002/002010.html
Size: 36899
Last Modified: Wed Dec 02 19:04:31 GMT-500 2015
Owner: used_stuff_cheap
ETag: 77d6d99a0e52e56a1081bd1c8adf80e6
Expiry Date: None
Expiration Rule: N/A

Each object has optional metadata



All Buckets / govdocs1 / files / 002

Name	Storage Class	Size	Last
002000.html	Standard	105.4 KB	Wed I
002001.html	Standard	8.2 KB	Wed I
002002.html	Standard	14.5 KB	Wed I
002003.html	Standard	35.6 KB	Wed I
002004.html	Standard	41.1 KB	Wed I
002005.html	Standard	45.7 KB	Wed I
002006.html	Standard	13.6 KB	Wed I
002007.html	Standard	642 bytes	Wed I
002008.html	Standard	60.5 KB	Wed I
002009.html	Standard	9.2 KB	Wed I
002010.html	Reduced Redundancy	36 KB	Wed I
002011.html	Standard	15.5 KB	Wed I
002012.html	Standard	31.9 KB	Wed I
002013.html	Standard	10 KB	Wed I
002014.pdf	Standard	2.1 MB	Wed I
002015.html	Standard	64.3 KB	Wed I
002016.html	Standard	3.1 KB	Wed I
002017.html	Standard	17.1 KB	Wed I
002018.html	Standard	3.2 KB	Wed I
002019.html	Standard	24.8 KB	Wed I
002020.html	Standard	5 KB	Wed I

Object: 002013.html

Bucket: govdocs1
Folder: 002
Name: 002013.html
Link: https://s3.amazonaws.com/govdocs1/files/002/002013.html
Size: 10281
Last Modified: Wed Dec 02 19:04:32 GMT-500 2015
Owner: used_stuff_cheap
ETag: 1d442e7e524223886636ff97792c3ce3
Expiry Date: None
Expiration Rule: N/A

▸ Details

▸ Permissions

▾ Metadata

Metadata is a set of name-value pairs. Learn more.

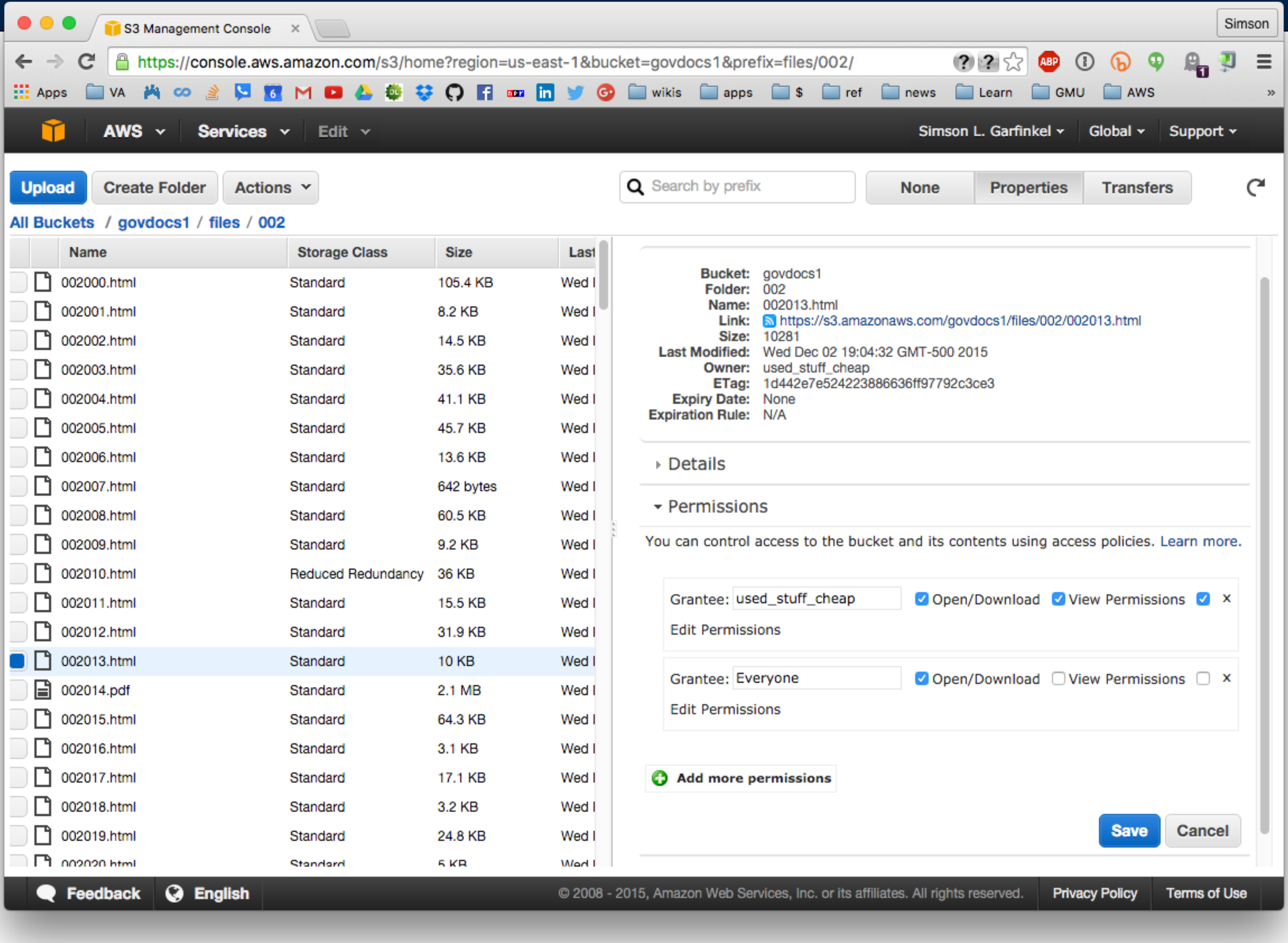
Key: Content-Type Value: application/octet-stream ×

Add more metadata   Remove selected metadata

Save   Cancel

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Permissions can be set per-bucket



All Buckets / govdocs1 / files / 002

Name	Storage Class	Size	Last
002000.html	Standard	105.4 KB	Wed I
002001.html	Standard	8.2 KB	Wed I
002002.html	Standard	14.5 KB	Wed I
002003.html	Standard	35.6 KB	Wed I
002004.html	Standard	41.1 KB	Wed I
002005.html	Standard	45.7 KB	Wed I
002006.html	Standard	13.6 KB	Wed I
002007.html	Standard	642 bytes	Wed I
002008.html	Standard	60.5 KB	Wed I
002009.html	Standard	9.2 KB	Wed I
002010.html	Reduced Redundancy	36 KB	Wed I
002011.html	Standard	15.5 KB	Wed I
002012.html	Standard	31.9 KB	Wed I
002013.html	Standard	10 KB	Wed I
002014.pdf	Standard	2.1 MB	Wed I
002015.html	Standard	64.3 KB	Wed I
002016.html	Standard	3.1 KB	Wed I
002017.html	Standard	17.1 KB	Wed I
002018.html	Standard	3.2 KB	Wed I
002019.html	Standard	24.8 KB	Wed I
002020.html	Standard	5 KB	Wed I

Bucket: govdocs1
Folder: 002
Name: 002013.html
Link: https://s3.amazonaws.com/govdocs1/files/002/002013.html
Size: 10281
Last Modified: Wed Dec 02 19:04:32 GMT-500 2015
Owner: used_stuff_cheap
ETag: 1d442e7e524223886636ff97792c3ce3
Expiry Date: None
Expiration Rule: N/A

▸ Details

▾ Permissions

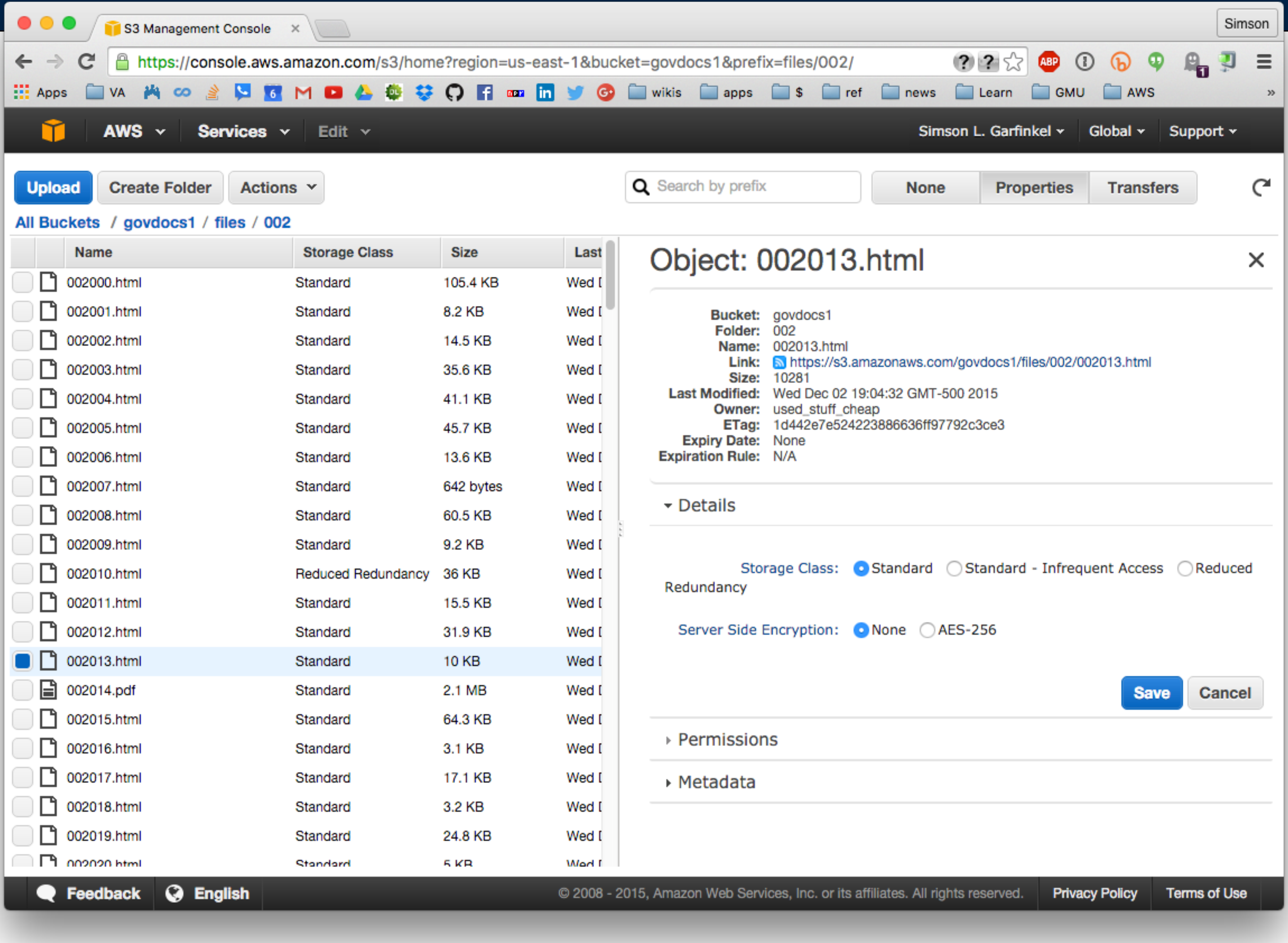
You can control access to the bucket and its contents using access policies. Learn more.

Grantee: used_stuff_cheap ☑ Open/Download ☑ View Permissions ☑ Edit Permissions ×

Grantee: Everyone ☑ Open/Download ☐ View Permissions ☐ Edit Permissions ×

Add more permissions
Save
Cancel
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Storage class can be set per-object or per-bucket



All Buckets / govdocs1 / files / 002

Name	Storage Class	Size	Last
002000.html	Standard	105.4 KB	Wed
002001.html	Standard	8.2 KB	Wed
002002.html	Standard	14.5 KB	Wed
002003.html	Standard	35.6 KB	Wed
002004.html	Standard	41.1 KB	Wed
002005.html	Standard	45.7 KB	Wed
002006.html	Standard	13.6 KB	Wed
002007.html	Standard	642 bytes	Wed
002008.html	Standard	60.5 KB	Wed
002009.html	Standard	9.2 KB	Wed
002010.html	Reduced Redundancy	36 KB	Wed
002011.html	Standard	15.5 KB	Wed
002012.html	Standard	31.9 KB	Wed
002013.html	Standard	10 KB	Wed
002014.pdf	Standard	2.1 MB	Wed
002015.html	Standard	64.3 KB	Wed
002016.html	Standard	3.1 KB	Wed
002017.html	Standard	17.1 KB	Wed
002018.html	Standard	3.2 KB	Wed
002019.html	Standard	24.8 KB	Wed
002020.html	Standard	5 KB	Wed

Object: 002013.html

Bucket:	govdocs1
Folder:	002
Name:	002013.html
Link:	https://s3.amazonaws.com/govdocs1/files/002/002013.html
Size:	10281
Last Modified:	Wed Dec 02 19:04:32 GMT-500 2015
Owner:	used_stuff_cheap
ETag:	1d442e7e524223886636ff97792c3ce3
Expiry Date:	None
Expiration Rule:	N/A

Details

Storage Class: ● Standard ○ Standard - Infrequent Access ○ Reduced Redundancy

Server Side Encryption: ● None ○ AES-256

Save Cancel

Permissions

Metadata

S3 Pricing — You pay for what you use (as of 2017-01-05)

AWS Free Usage Tier — 1 year after you sign up

- 5GB of S3 Storage

Each month:

- 20,000 GET requests
- 2,000 PUT requests
- 15GB of data transfer free

- http://aws.amazon.com/s3/reduced-redundancy/

Pricing: S3 vs. EBS (as of 2015-12-15)

Price for 1TB

	Standard Storage	Standard - Infrequent Access Storage †	Glacier Storage
First 50 TB / month	$0.023 per GB	$0.0125 per GB	$0.004 per GB
Next 450 TB / month	$0.022 per GB	$0.0125 per GB	$0.004 per GB
Next 500 TB / month	$0.021 per GB	$0.0125 per GB	$0.004 per GB
PUT, COPY, POST, LIST	$0.01 per 1,000	$0.01 per 1,000	$0.05 per 1,000
GET requests	$0.01 per 10,000	$0.01 per 1,000	1-5 min: $0.03 per GB 3-5 hours: $0.01 per GB 5-12 hours: $.025 per 1000
DEL	Free	Free	‡

- http://aws.amazon.com/s3/pricing/
- http://aws.amazon.com/s3/reduced-redundancy/

‡ charged for deleting objects less than 3 months old

Upload files to S3 with Python: use “boto”

Simple program to upload the file ‘000.zip’
to s3://simsong/govdocs1/zipfiles/000.zip:

```
#!/usr/bin/env python
# https://aws.amazon.com/articles/Python/3998

import boto
s3 = boto.connect_s3()
bucket = s3.get_bucket('govdocs1')
key = bucket.new_key('zipfiles/z1')
key.set_contents_from_filename(‘000.zip')
key.set_acl(‘public-read')
```

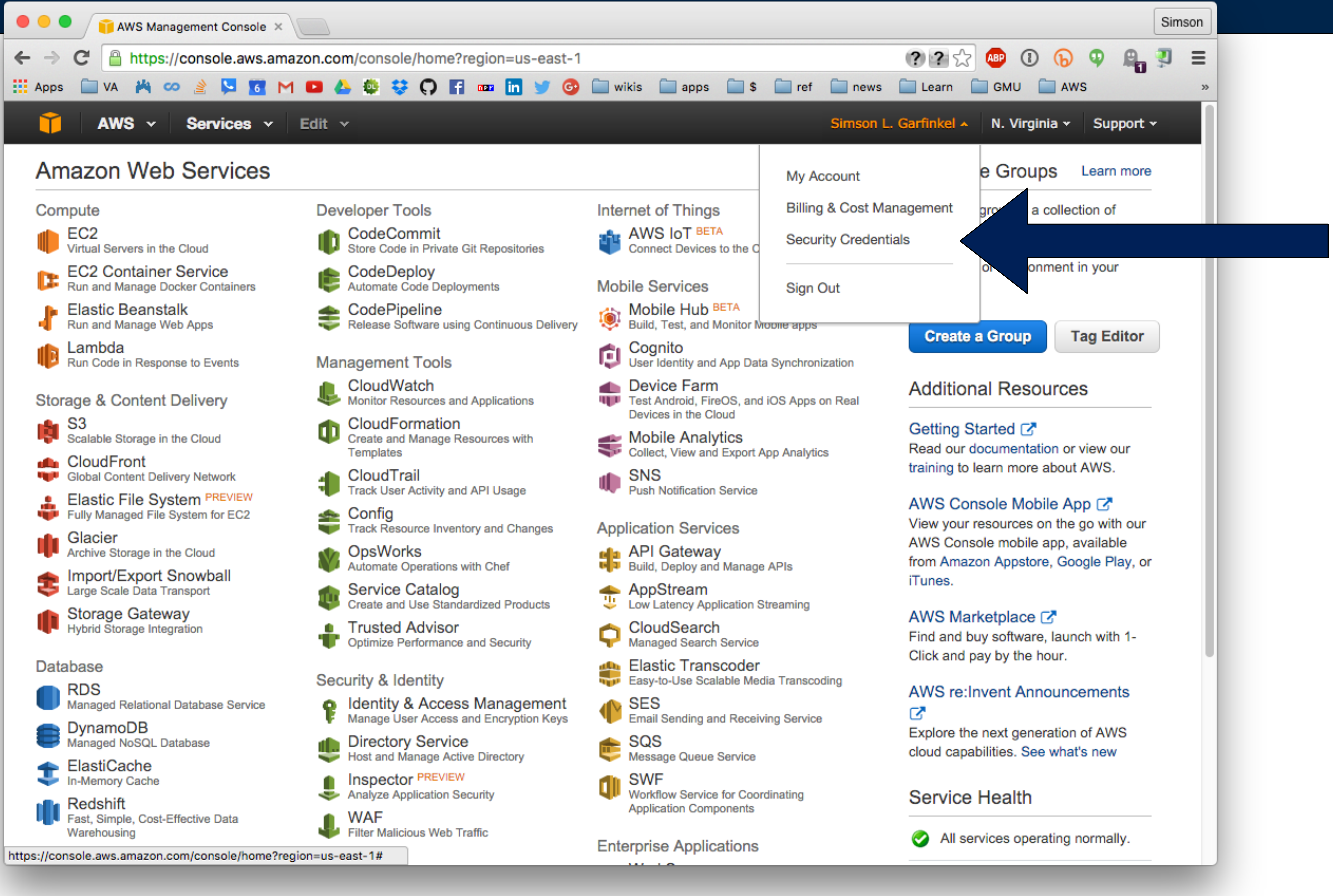
This program doesn’t work as-is….

The uploader doesn’t have permission to upload to S3.

```
$ python uploader.py
Traceback (most recent call last):
  File "uploader.py", line 8, in <module>
    key.set_contents_from_filename('uploader.py')
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 1362, in set_contents_from_filename
    encrypt_key=encrypt_key)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 1293, in set_contents_from_file
    chunked_transfer=chunked_transfer, size=size)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 750, in send_file
    chunked_transfer=chunked_transfer, size=size)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 951, in _send_file_internal
    query_args=query_args
  File "/usr/lib/python2.7/dist-packages/boto/s3/connection.py", line 664, in make_request
    retry_handler=retry_handler
  File "/usr/lib/python2.7/dist-packages/boto/connection.py", line 1071, in make_request
    retry_handler=retry_handler)
  File "/usr/lib/python2.7/dist-packages/boto/connection.py", line 940, in _mexe
    request.body, request.headers)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 884, in sender
    response.status, response.reason, body)
boto.exception.S3ResponseError: S3ResponseError: 403 Forbidden
<?xml version="1.0" encoding="UTF-8"?>
<Error><Code>AccessDenied</Code><Message>Access Denied</Message><RequestId>3C2C61650BFDDC0D</
RequestId><HostId>u7DYAYhQVUiaygmnScU0cmozaU8kRofETXoiH00yLC/8jYqcS4aNSfWRJaSWDu0GeKRFyzizQ28=</HostId></Error>
$
```

Boto requires AWS authentication

Create/get your security credentials



AWS Management Console

https://console.aws.amazon.com/console/home?region=us-east-1

My Account

Billing & Cost Management

Security Credentials

Sign Out

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 BETA – Connect Devices to the C

Mobile Services

- Mobile Hub BETA – 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 – Message Queue Service
- SWF – Workflow Service for Coordinating Application Components

Enterprise Applications

e Groups Learn more

gro a collection of

o onment in your

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.

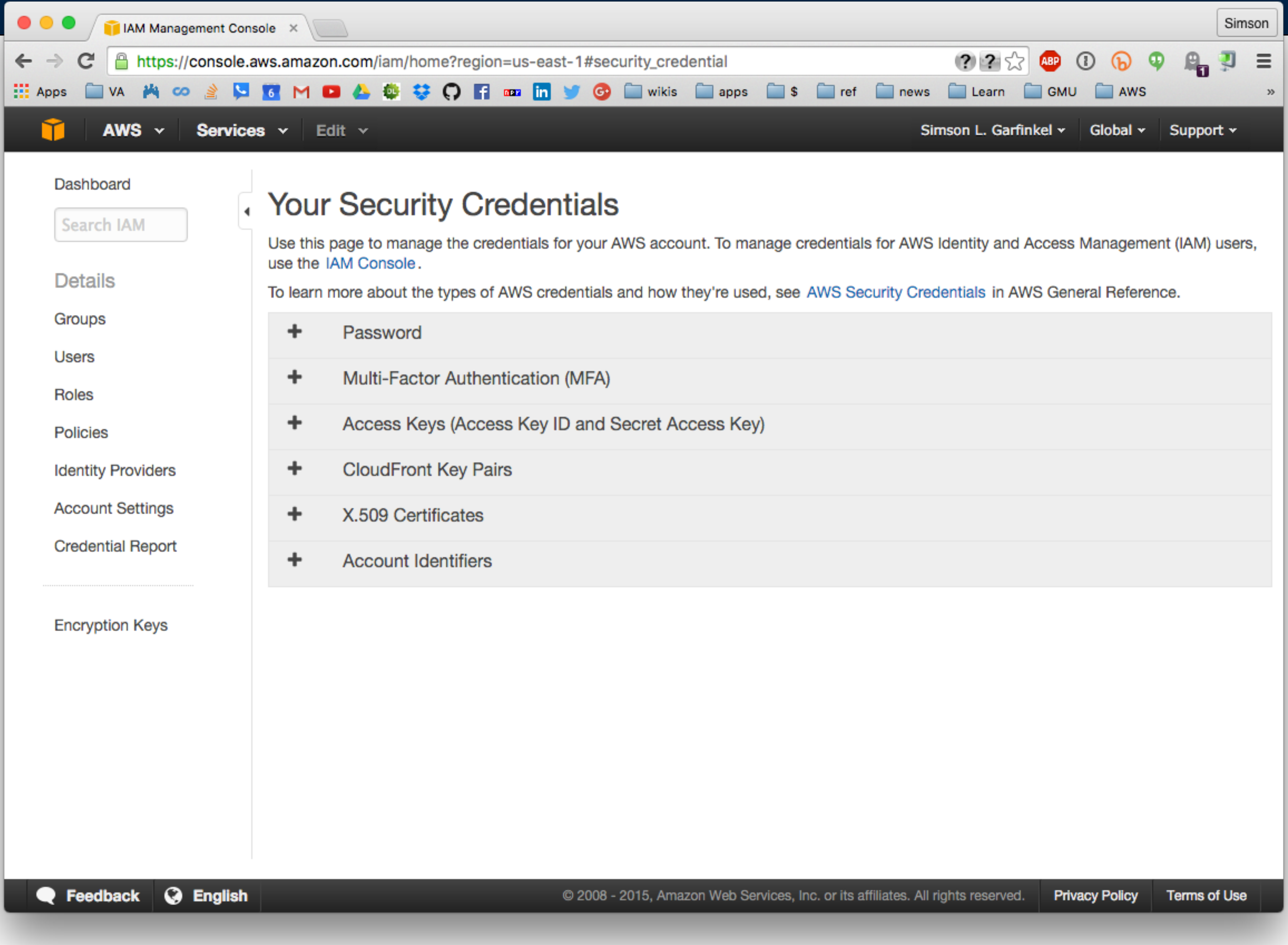
AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.

AWS re:Invent Announcements
Explore the next generation of AWS cloud capabilities. See what's new

Service Health

All services operating normally.

https://console.aws.amazon.com/console/home?region=us-east-1#



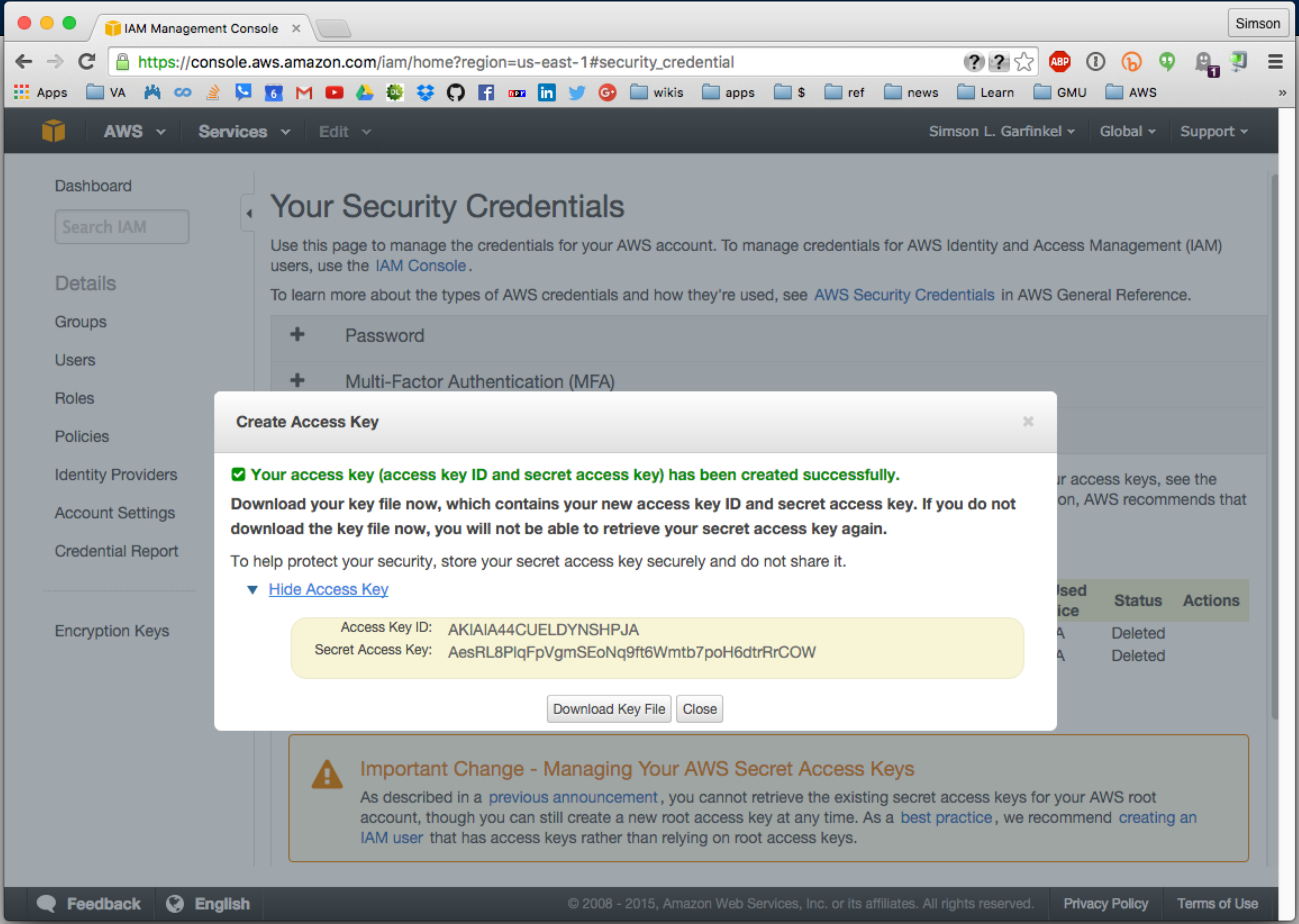
Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

- ✚ Password
- ✚ Multi-Factor Authentication (MFA)
- ✚ Access Keys (Access Key ID and Secret Access Key)
- ✚ CloudFront Key Pairs
- ✚ X.509 Certificates
- ✚ Account Identifiers

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

ur access keys, see the on, AWS recommends that

Create Access Key

☑ **Your access key (access key ID and secret access key) has been created successfully.**

Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.

To help protect your security, store your secret access key securely and do not share it.

▼ Hide Access Key

Access Key ID:	AKIAIA44CUELDYNSHPJA
Secret Access Key:	AesRL8PlqFpVgmSEoNq9ft6Wmtb7poH6dtrRrCOW

Important Change - Managing Your AWS Secret Access Keys

As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

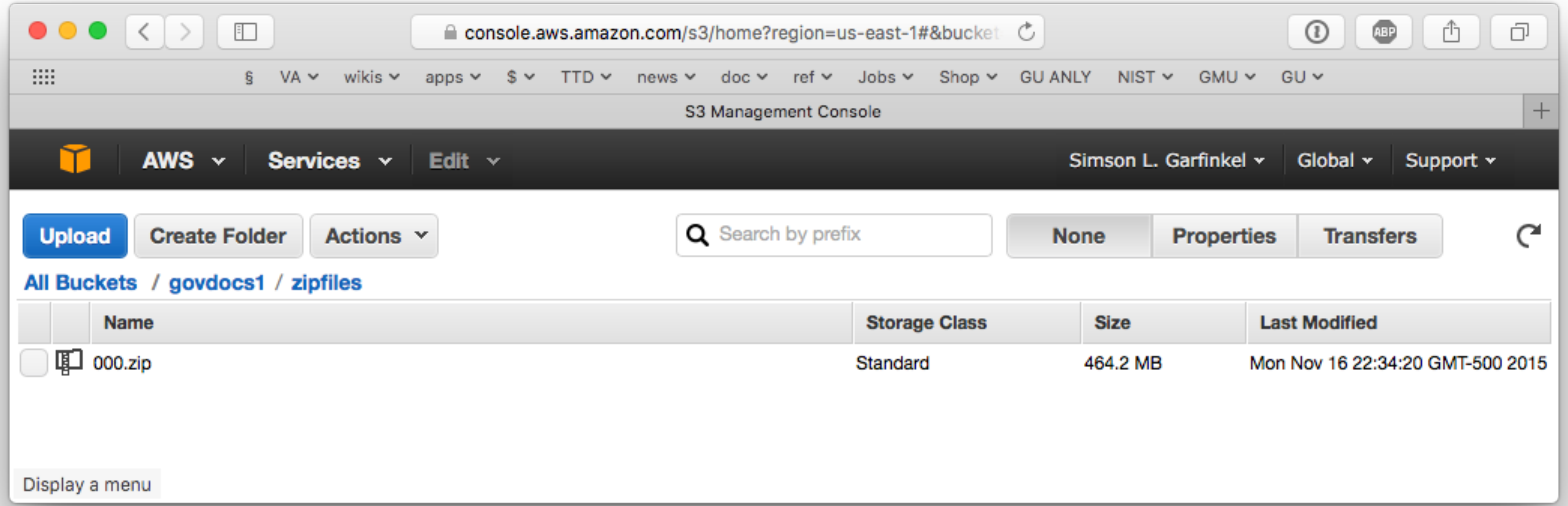
Boto requires AWS authentication
Store in the file ~/.boto

```
$ cat ~/.boto
[Credentials]
aws_access_key_id = AKIAJNMBJEXXYO6USPHD
aws_secret_access_key = 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
$
```

Now let's run it:

```
$ python uploader.py
$
```

And we can verify it's there:



Name	Storage Class	Size	Last Modified
000.zip	Standard	464.2 MB	Mon Nov 16 22:34:20 GMT-500 2015

S3 Authentication

Access to S3 is authenticated with two secret keys:

- AWS Access Key ID — 20 Character String
  - *e.g. AKIAIOSFODNN7EXAMPLE*
- AWS Secret Key — 40-character string
  - *e.g. wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*

Keys can be created:

- For primary user account
- For IAM accounts
- For individual applications

Public key cryptography is not used!

- Credentials must be sent over SSL

Uploading to S3

Simple program to upload the file ‘000.zip’ to s3://simsong/govdocs1/zipfiles/000.zip:

```
#!/usr/bin/env python
# https://aws.amazon.com/articles/Python/3998

import boto
s3 = boto.connect_s3()
bucket = s3.get_bucket('govdocs1')
key = bucket.new_key('zipfiles/z1')
key.set_contents_from_filename(‘000.zip')
key.set_acl(‘public-read')
```

Permission errors on S3 can be obscure

This means you don't have permission to upload:

```
Uploading s3://govdocs1/400.zip
Traceback (most recent call last):
  File "uploader.py", line 75, in <module>
    results.append(copy(i))
  File "uploader.py", line 41, in copy
    key.set_contents_from_filename(fname)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 1362, in set_contents_from_filename
    encrypt_key=encrypt_key)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 1293, in set_contents_from_file
    chunked_transfer=chunked_transfer, size=size)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 750, in send_file
    chunked_transfer=chunked_transfer, size=size)
  File "/usr/lib/python2.7/dist-packages/boto/s3/key.py", line 951, in _send_file_internal
    query_args=query_args
  File "/usr/lib/python2.7/dist-packages/boto/s3/connection.py", line 664, in make_request
    retry_handler=retry_handler
  File "/usr/lib/python2.7/dist-packages/boto/connection.py", line 1071, in make_request
    retry_handler=retry_handler)
  File "/usr/lib/python2.7/dist-packages/boto/connection.py", line 1030, in _mexe
    raise ex
socket.error: [Errno 104] Connection reset by peer
```

S3 “Requester Pays”

Normally the bucket owner pays for access fees.

With Requester Pays, the requester pays.

- No anonymous access.
- No charge to download within EC2
- No BitTorrent or SOAP

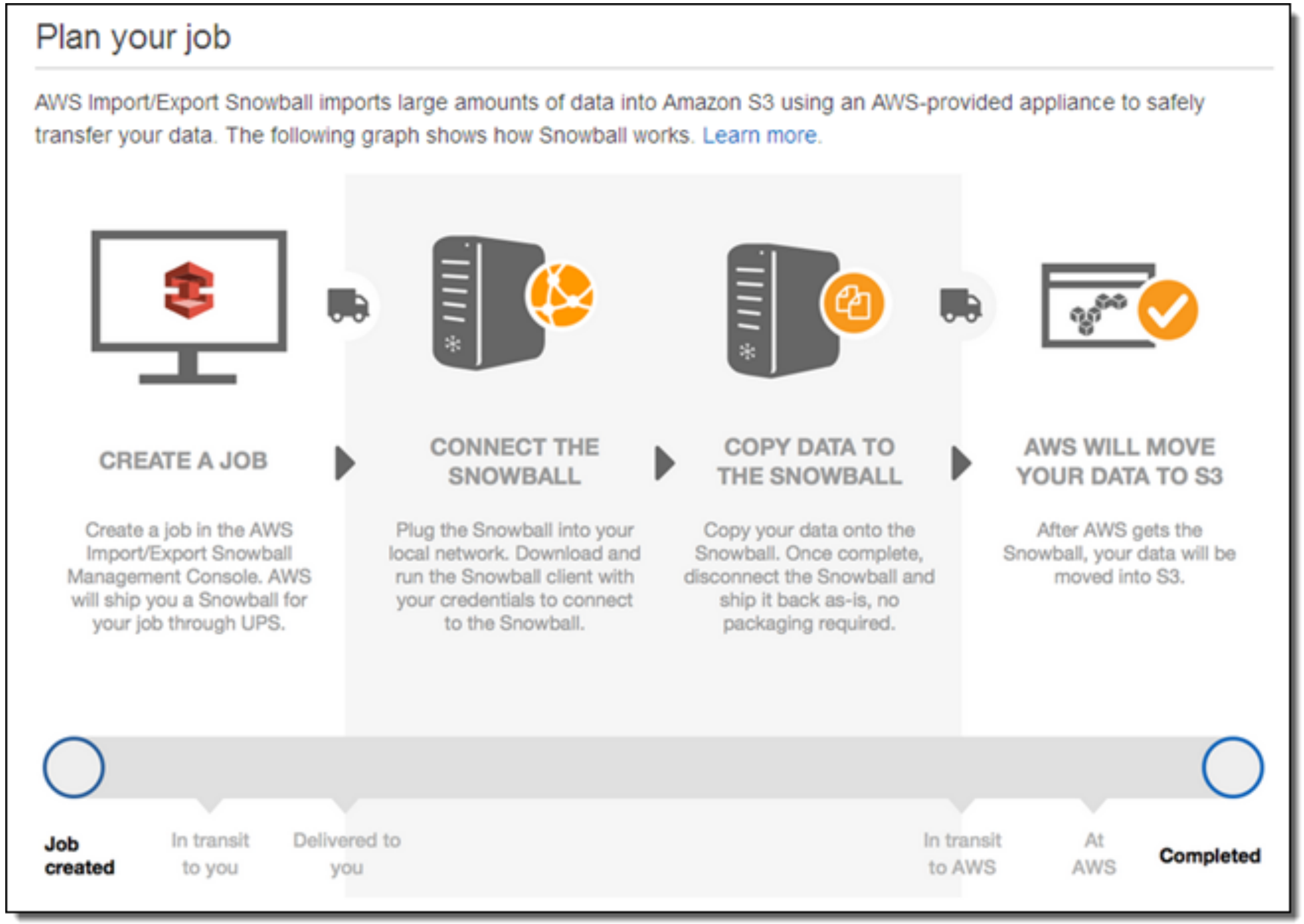
“DevPay” lets you sell your content

- Requestors are charged to access data.
- Can be combined with Requester Pays
  - https://docs.aws.amazon.com/AmazonS3/latest/dev/RequesterPaysBuckets.html
  - http://docs.aws.amazon.com/AmazonDevPay/latest/DevPayDeveloperGuide/S3RequesterPays.html

Amazon Snowball

- https://aws.amazon.com/blogs/aws/aws-importexport-snowball-transfer-1-petabyte-per-week-using-amazon-owned-storage-appliances/

Amazon Snowball



Plan your job

AWS Import/Export Snowball imports large amounts of data into Amazon S3 using an AWS-provided appliance to safely transfer your data. The following graph shows how Snowball works. Learn more.

CREATE A JOB

Create a job in the AWS Import/Export Snowball Management Console. AWS will ship you a Snowball for your job through UPS.

CONNECT THE SNOWBALL

Plug the Snowball into your local network. Download and run the Snowball client with your credentials to connect to the Snowball.

COPY DATA TO THE SNOWBALL

Copy your data onto the Snowball. Once complete, disconnect the Snowball and ship it back as-is, no packaging required.

AWS WILL MOVE YOUR DATA TO S3

After AWS gets the Snowball, your data will be moved into S3.

- https://aws.amazon.com/blogs/aws/aws-importexport-snowball-transfer-1-petabyte-per-week-using-amazon-owned-storage-appliances/

Backup Slides

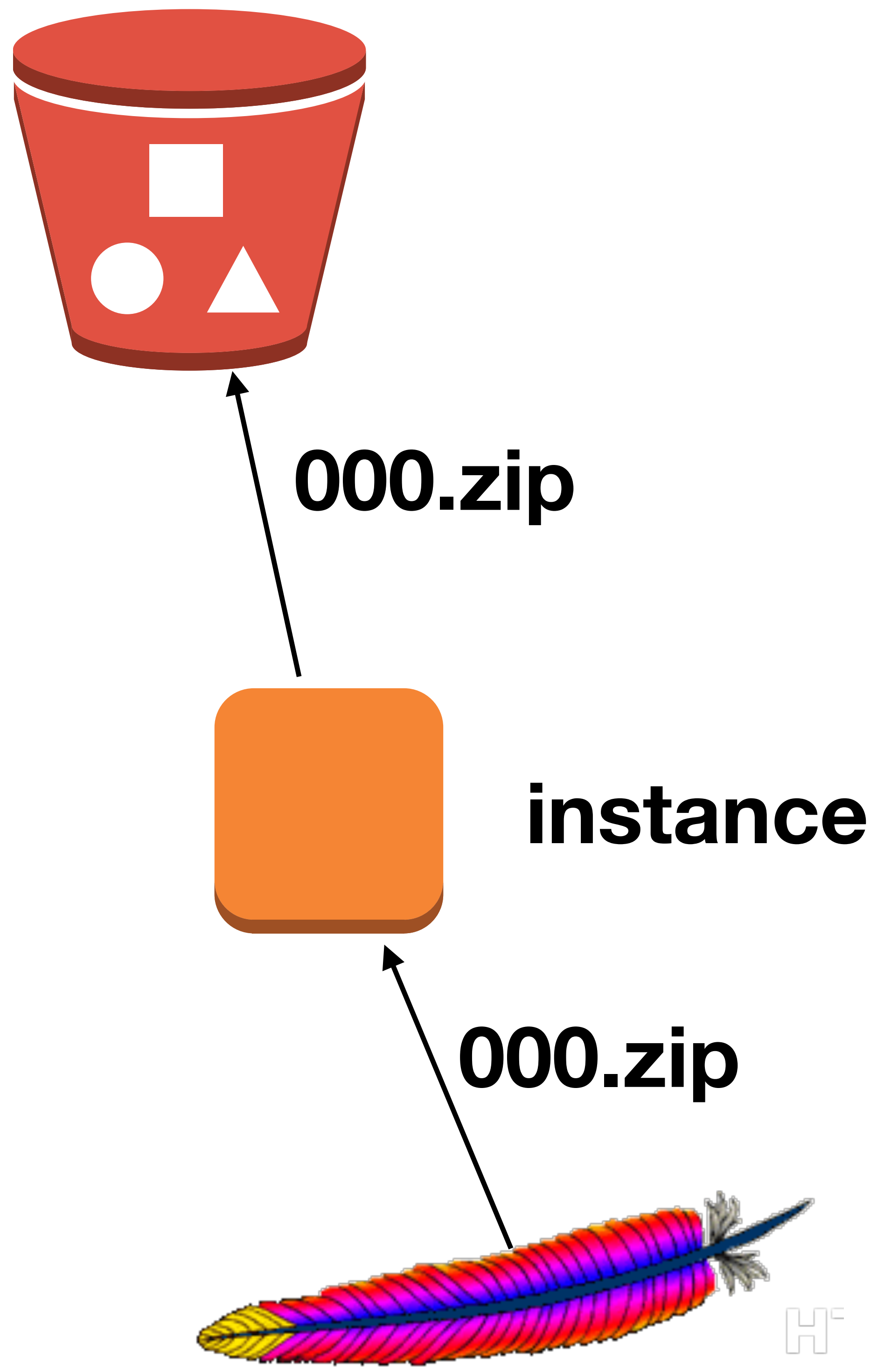
Uploading data to S3

S3 task: Move 1000 files from a web server to Amazon S3

First approach:

- For each file N:
  - —*Get the file from web server.*
  - —*Send the file to S3*

How would you do this?



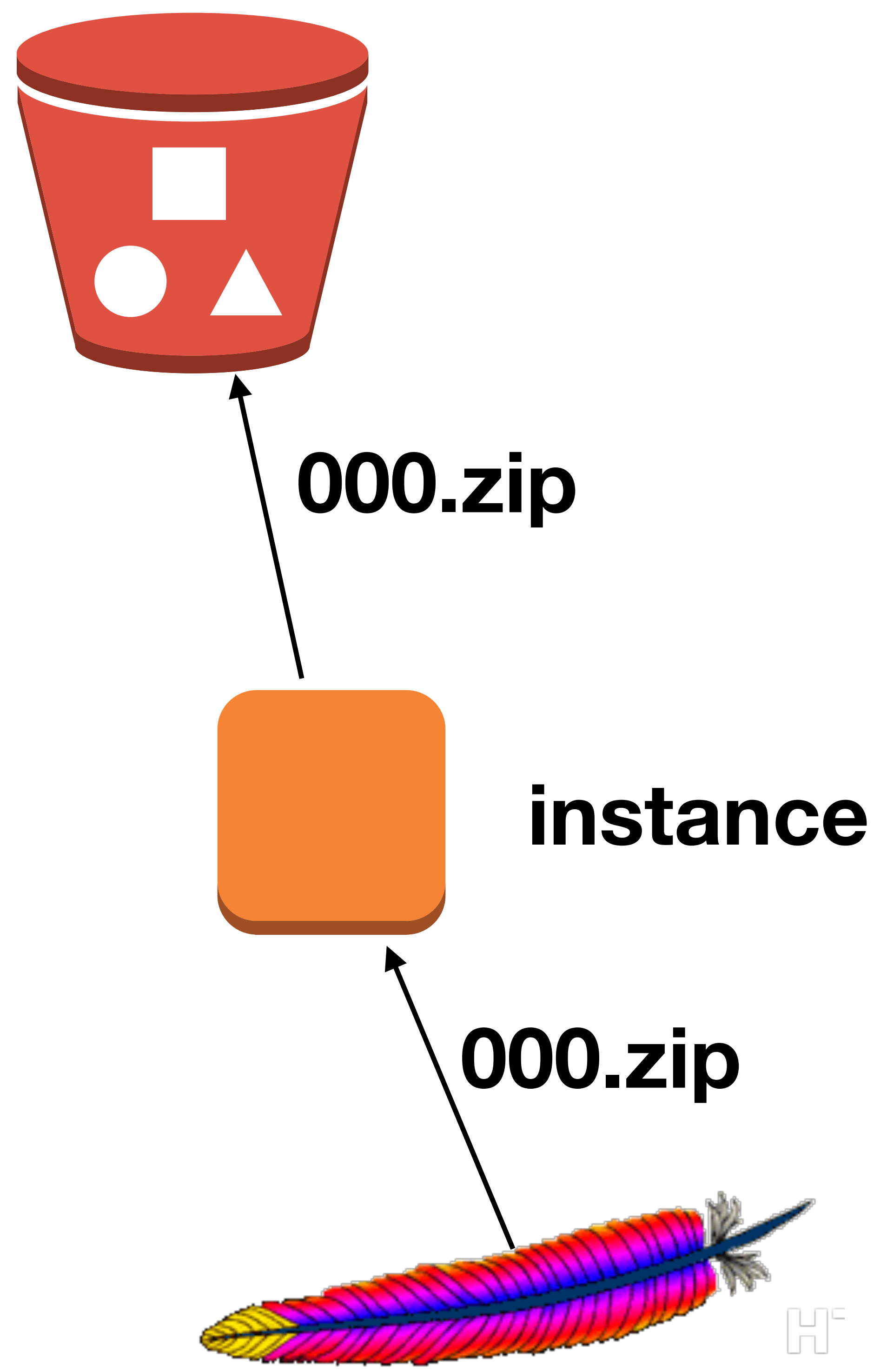
S3 task: Move 1000 files from a web server to Amazon S3

First approach:

- For each file N:
  - *Get the file from web server.*
  - *Send the file to S3*

Potential problems:

- Download might be interrupted.
- Upload might be interrupted.
- Server might crash.



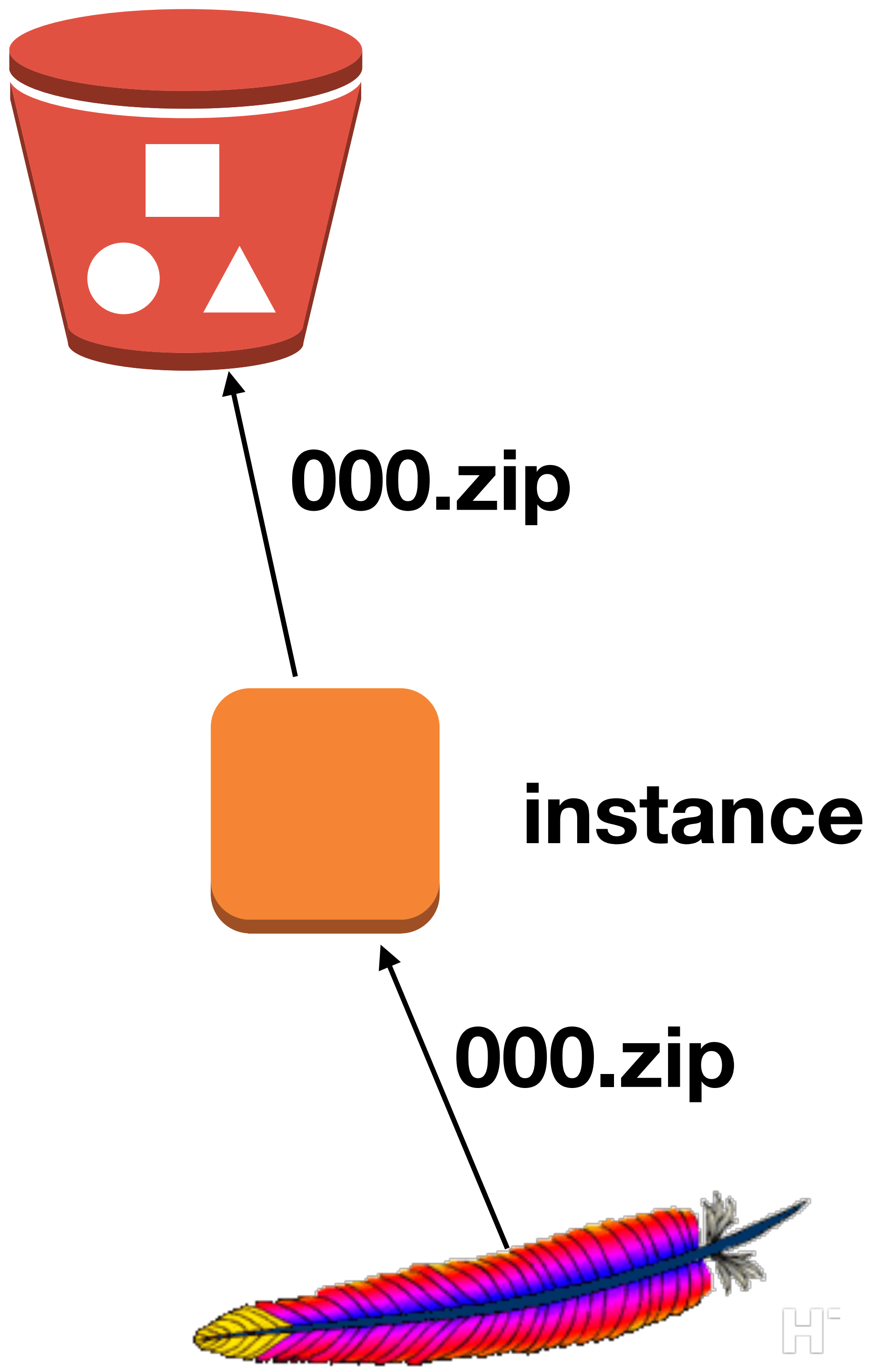
S3 task: Move 1000 files from a web server to Amazon S3

First approach:

- For each file N:
  - *Get the file from web server.*
  - *Send the file to S3*

Revised approach:

- For each file N:
  - *Get the size of file N from the WWW server*
  - *If file N is not on S3, or if it is the wrong size:*
  - *If file N is not on the instance, or is the wrong size:*
  - *Download the file from the WWW server to the instance*
  - *Upload the file to S3*



Notice that this is "indempotent"

- Tolerant to failure and restarting from beginning at any point.
- Tolerant to being run a number of times.

copy(n): Download the zipfile and upload the parts

```python
def copy(n):
    fname = "%03i.zip" % n

    # get the remove file length
    url = 'http://digitalcorpora.org/corp/files/govdocs1/zipfiles/'+fname
    u = urllib2.urlopen(url)
    meta = u.info()
    file_size = int(meta.getheaders("Content-Length")[0])

    # get the key in the bucket
    key = bucket.lookup('zipfiles/'+fname)
    if key and key.exists and key.size==file_size:
        print("{} exists and is correct size ({:,}B)".format(fname,file_size))
        return (fname,0,0)
    if not key:
        key = bucket.new_key('zipfiles/'+fname)

    # Download the file if we don't have it
   if os.path.exists(fname)==False or os.path.getsize(fname)!=file_size:
        print("Downloading {}".format(url))
        block_sz = 65536
        with open(fname,"wb") as f:
            while True:
                buffer = u.read(block_sz)
                if not buffer:
                    break
                f.write(buffer)

    # Upload the file
    print("Uploading s3://{}/{}".format(bucket_name,fname))
    key.set_contents_from_filename(fname)
    key.set_acl('public-read')
    print("Uploaded {} {:,}B in {}s".format(fname,file_size,t1-t0))

    # Finally remove the uploaded file
    os.unlink(fname)
    return (fname,total_time, file_size)
```

note: total_time calculation removed

First driver program

```
if __name__=="__main__":
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('num',type=int,nargs='+')
   args = parser.parse_args()
    print(args.num)
    if len(args.num)==1:
        a = args.num[0]
        b = args.num[0]
    else:
        (a,b) = args.num[0:2]
        print(a,b)
    total_t  = 0
    total_sz = 0

    results = []
    start_time = time.time()
    for i in range(a,b+1):
        results.append(copy(i))

    end_time = time.time()
    real_time = end_time - start_time
    total_time = sum([r[1] for r in results])
    total_bytes = sum([r[2] for r in results])
    if total_time==0:
        print("nothing uploaded")
    else:
        print("Total uploaded {:,}MB in {}s, {:,}MB/sec".format(total_bytes/1E6,total_time,total_bytes/total_time/1E6))
        print("Effective upload: {:,}MB/sec in {} sec".format(total_bytes/real_time/1E6,real_time))
```

Run single-threaded on a t2.micro …
vCPU=1, CPU Credits/hour=6, Mem=1GiB, EBS-Only, Low Net

```
files/028/028753.txt uploaded
028/028754.txt
files/028/028754.txt uploaded
028/028755.txt
files/028/028755.txt uploaded
028/028756.txt
files/028/028756.txt uploaded
028/028757.txt
files/028/028757.txt uploaded
028/028758.txt
Traceback (most recent call last):
  File "govdocs.py", line 56, in <module>
    putzipparts(tfn)
  File "govdocs.py", line 37, in putzipparts
    data = z.open(zname,"r").read()
  File "/usr/lib64/python2.7/zipfile.py", line 630, in read
    data = self.read1(n)
  File "/usr/lib64/python2.7/zipfile.py", line 684, in read1
    max(n - len_readbuffer, self.MIN_READ_SIZE)
MemoryError
$
```

Total time to upload 28 files ≈ 10 minutes (until I interrupted it)

To upload 1000 files would take: 28÷10 × 1000 ≈ 2800 minutes ≈ 2 days

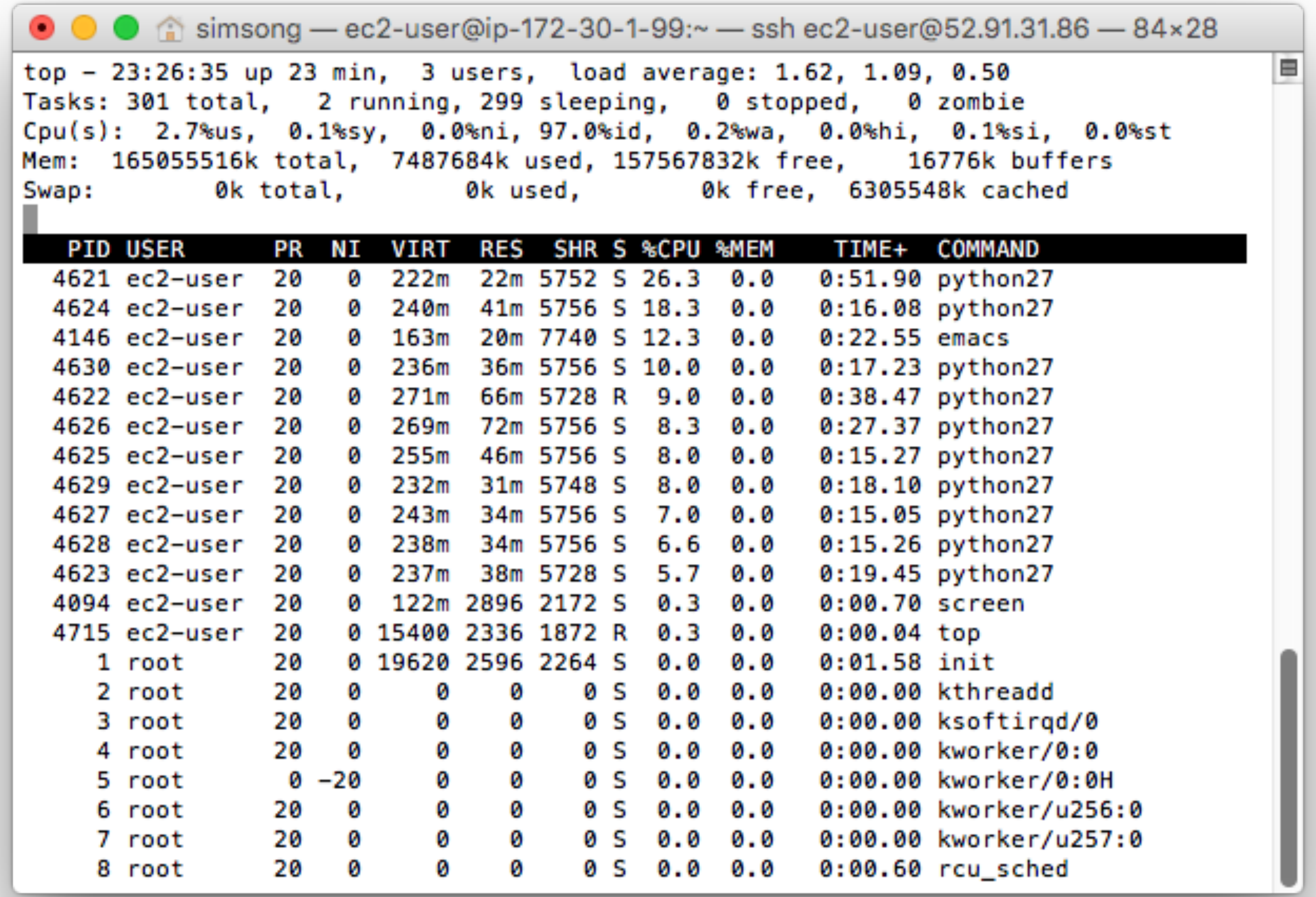
Modified driver program supports multithreading with multiprocessing Pool

```
if __name__=="__main__":
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('num',type=int,nargs='+')
    parser.add_argument('--multi',type=int)
    args = parser.parse_args()
    print(args.num)
    if len(args.num)==1:
        a = args.num[0]
        b = args.num[0]
    else:
        (a,b) = args.num[0:2]
        print(a,b)
    total_t  = 0
    total_sz = 0

    results = []
    start_time = time.time()
    if args.multi:
        from multiprocessing import Pool
        print("range: {} to {}".format(a,b+1))
        p = Pool(args.multi)
        results = p.map(copy,range(a,b+1))
    else:
        for i in range(a,b+1):
            results.append(copy(i))

    end_time = time.time()
    real_time = end_time - start_time
    total_time = sum([r[1] for r in results])
    total_bytes = sum([r[2] for r in results])
    if total_time==0:
        print("nothing uploaded")
    else:
        print("Total uploaded {:,}MB in {}s, {:,}MB/sec".format(total_bytes/1E6,total_time,total_bytes/total_time/1E6))
        print("Effective upload: {:,}MB/sec in {} sec".format(total_bytes/real_time/1E6,real_time))
```

multithreaded

Run with 1-6 threads to upload ZIP files in 50-file batches

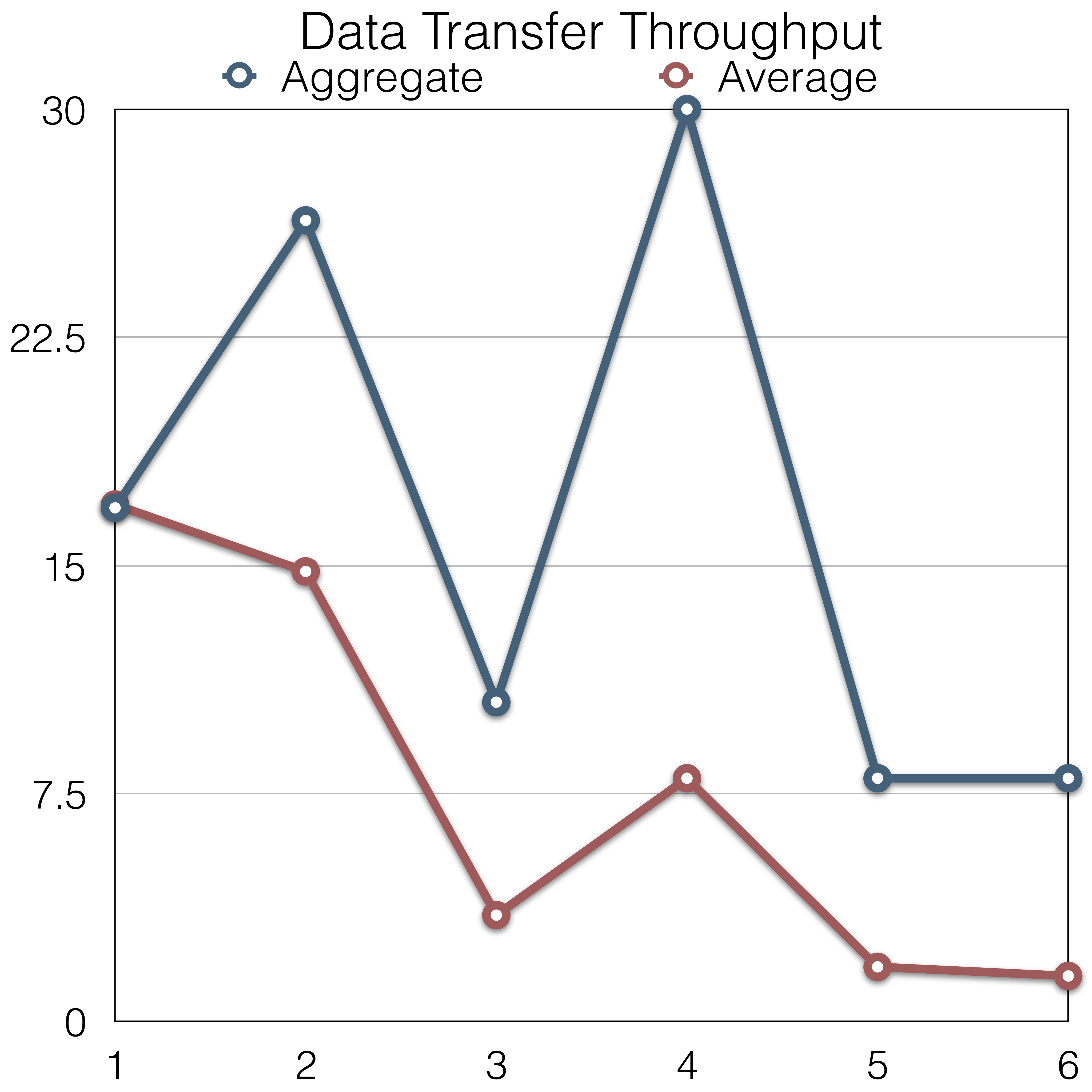
Verified multi threading:


simsong — ec2-user@ip-172-30-1-99:~ — ssh ec2-user@52.91.31.86 — 84×28

```
top - 23:26:35 up 23 min,  3 users,  load average: 1.62, 1.09, 0.50
Tasks: 301 total,   2 running, 299 sleeping,   0 stopped,   0 zombie
Cpu(s):  2.7%us,  0.1%sy,  0.0%ni, 97.0%id,  0.2%wa,  0.0%hi,  0.1%si,  0.0%st
Mem:  165055516k total,  7487684k used, 157567832k free,    16776k buffers
Swap:        0k total,        0k used,        0k free,  6305548k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 4621 ec2-user  20   0  222m  22m 5752 S 26.3  0.0   0:51.90 python27
 4624 ec2-user  20   0  240m  41m 5756 S 18.3  0.0   0:16.08 python27
 4146 ec2-user  20   0  163m  20m 7740 S 12.3  0.0   0:22.55 emacs
 4630 ec2-user  20   0  236m  36m 5756 S 10.0  0.0   0:17.23 python27
 4622 ec2-user  20   0  271m  66m 5728 R  9.0  0.0   0:38.47 python27
 4626 ec2-user  20   0  269m  72m 5756 S  8.3  0.0   0:27.37 python27
 4625 ec2-user  20   0  255m  46m 5756 S  8.0  0.0   0:15.27 python27
 4629 ec2-user  20   0  232m  31m 5748 S  8.0  0.0   0:18.10 python27
 4627 ec2-user  20   0  243m  34m 5756 S  7.0  0.0   0:15.05 python27
 4628 ec2-user  20   0  238m  34m 5756 S  6.6  0.0   0:15.26 python27
 4623 ec2-user  20   0  237m  38m 5728 S  5.7  0.0   0:19.45 python27
 4094 ec2-user  20   0  122m 2896 2172 S  0.3  0.0   0:00.70 screen
 4715 ec2-user  20   0 15400 2336 1872 R  0.3  0.0   0:00.04 top
    1 root      20   0 19620 2596 2264 S  0.0  0.0   0:01.58 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/0:0
    5 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/u256:0
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kworker/u257:0
    8 root      20   0     0    0    0 S  0.0  0.0   0:00.60 rcu_sched
```

Performance improved from 1 to 2 threads, then decreased. (contention between threads.)

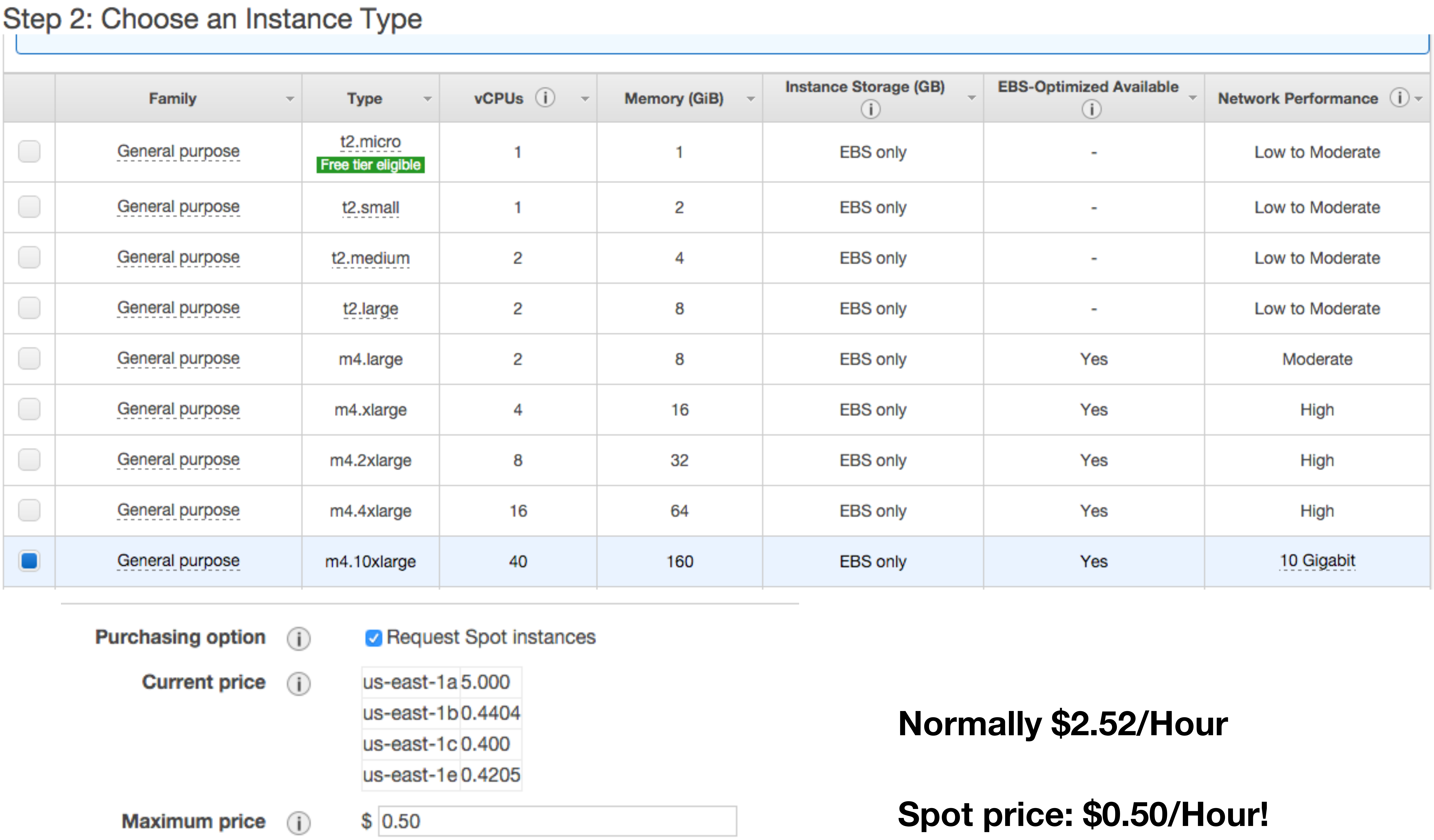
		Per Thread		Agregate	
Threads	Data Uploaded	Total Clock Time Σ(threads)	Avg throughput per thread	Wall Clock Time	Aggregate Througput
1	18GB	1087	17.0 MB/s	1087	16.89 MB/s
2	16 GB	1288	14.8 MB/s	644	26.34 MB/s
3	18 GB	1725	3.5 MB/s	575	10.5 MB/s
4	17 GB	2244	8 MB/s	561	30 MB/s
5	17 GB	9364	1.8 MB/s	1951	8 MB/s
6	17 GB	10,741	1.5 MB/s	1999	8 MB/s



Many core files.
Some of the sub processes were crashing—lack of memory!

```
$ ls -l
total 1127728
drwx------  4 ec2-user ec2-user     4096 Dec  2 02:00 anly502/
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:33 core.27872
-rw-------  1 ec2-user ec2-user  1404928 Dec  5 04:33 core.27880
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:34 core.27942
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:34 core.27998
-rw-------  1 ec2-user ec2-user  1400832 Dec  5 04:38 core.28025
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:39 core.28078
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:49 core.28138
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:49 core.28193
-rw-------  1 ec2-user ec2-user  1400832 Dec  5 04:49 core.28208
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:50 core.28266
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:50 core.28326
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:51 core.28381
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:52 core.28515
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:53 core.28607
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:53 core.28662
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:55 core.28715
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:55 core.28771
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:55 core.28824
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:58 core.28877
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:59 core.28932
-rw-------  1 ec2-user ec2-user   573440 Dec  5 05:00 core.28989
-rw-------  1 ec2-user ec2-user   573440 Dec  5 13:31 core.29694
-rw-------  1 ec2-user ec2-user   573440 Dec  5 16:05 core.30033
-rw-------  1 ec2-user ec2-user  1400832 Dec  5 18:04 core.30219
-rw-------  1 ec2-user ec2-user  1400832 Dec  5 18:04 core.30223
-rw-------  1 ec2-user ec2-user   573440 Dec  5 18:26 core.30427
-rw-------  1 ec2-user ec2-user   573440 Dec  2 03:18 core.3437
-rw-------  1 ec2-user ec2-user   573440 Dec  2 23:59 core.5249
-rw-------  1 ec2-user ec2-user   573440 Dec  3 00:04 core.5331
-rw-------  1 ec2-user ec2-user   573440 Dec  4 03:22 core.8704
-rw-------  1 ec2-user ec2-user   573440 Dec  5 04:01 core.936
```

Spin up a high capacity machine.



Step 2: Choose an Instance Type

	Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ
☐	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
☐	General purpose	m4.large	2	8	EBS only	Yes	Moderate
☐	General purpose	m4.xlarge	4	16	EBS only	Yes	High
☐	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
☐	General purpose	m4.4xlarge	16	64	EBS only	Yes	High
■	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit

Purchasing option ⓘ ☑ Request Spot instances

Current price ⓘ

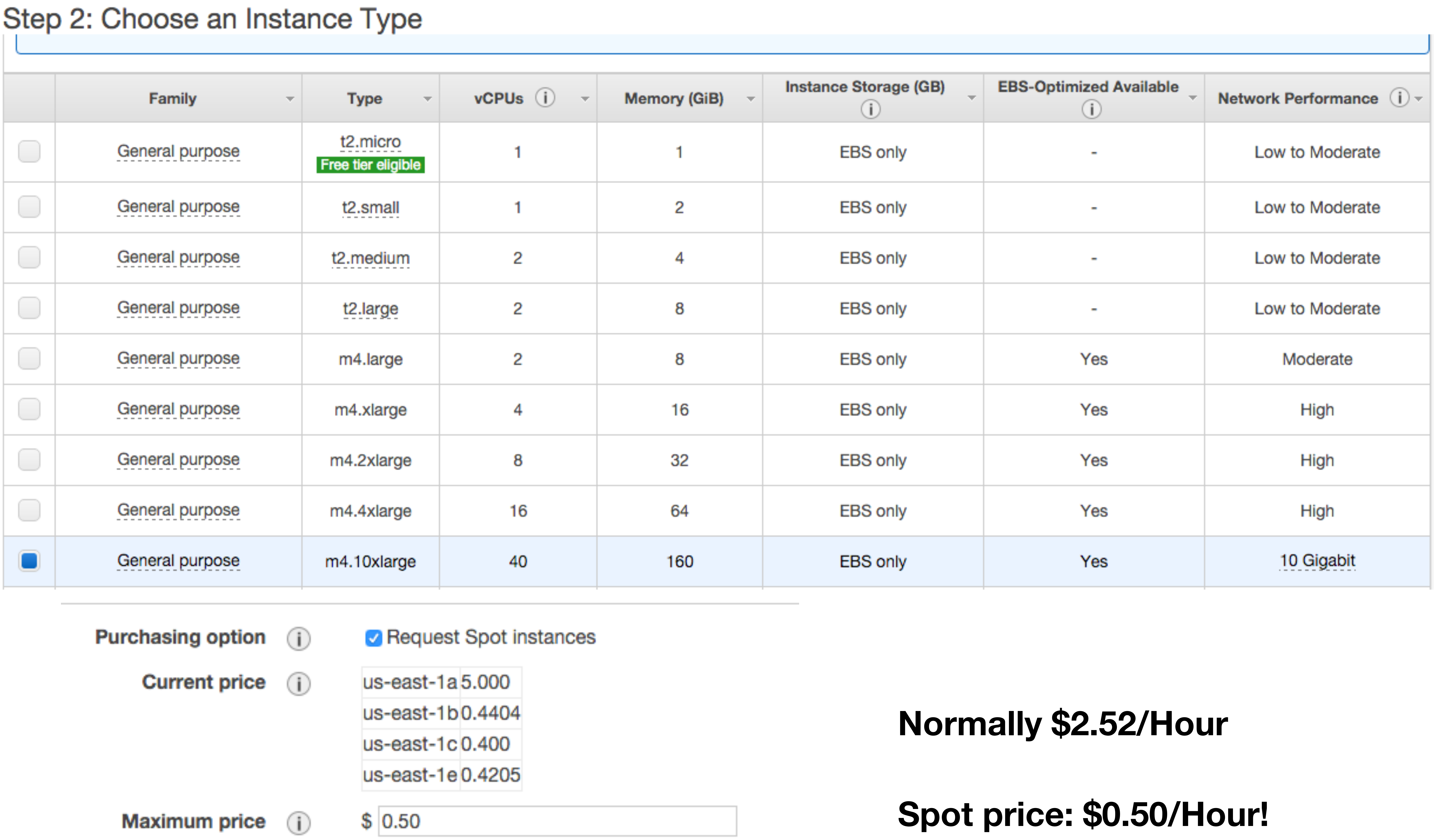
us-east-1a	5.000
us-east-1b	0.4404
us-east-1c	0.400
us-east-1e	0.4205

Maximum price ⓘ $ 0.50

Normally $2.52/Hour

Spot price: $0.50/Hour!

Spin up a high capacity machine.



Step 2: Choose an Instance Type

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate
□	General purpose	m4.xlarge	4	16	EBS only	Yes	High
□	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
□	General purpose	m4.4xlarge	16	64	EBS only	Yes	High
■	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit

Purchasing option ☑ Request Spot instances

Current price

us-east-1a	5.000
us-east-1b	0.4404
us-east-1c	0.400
us-east-1e	0.4205

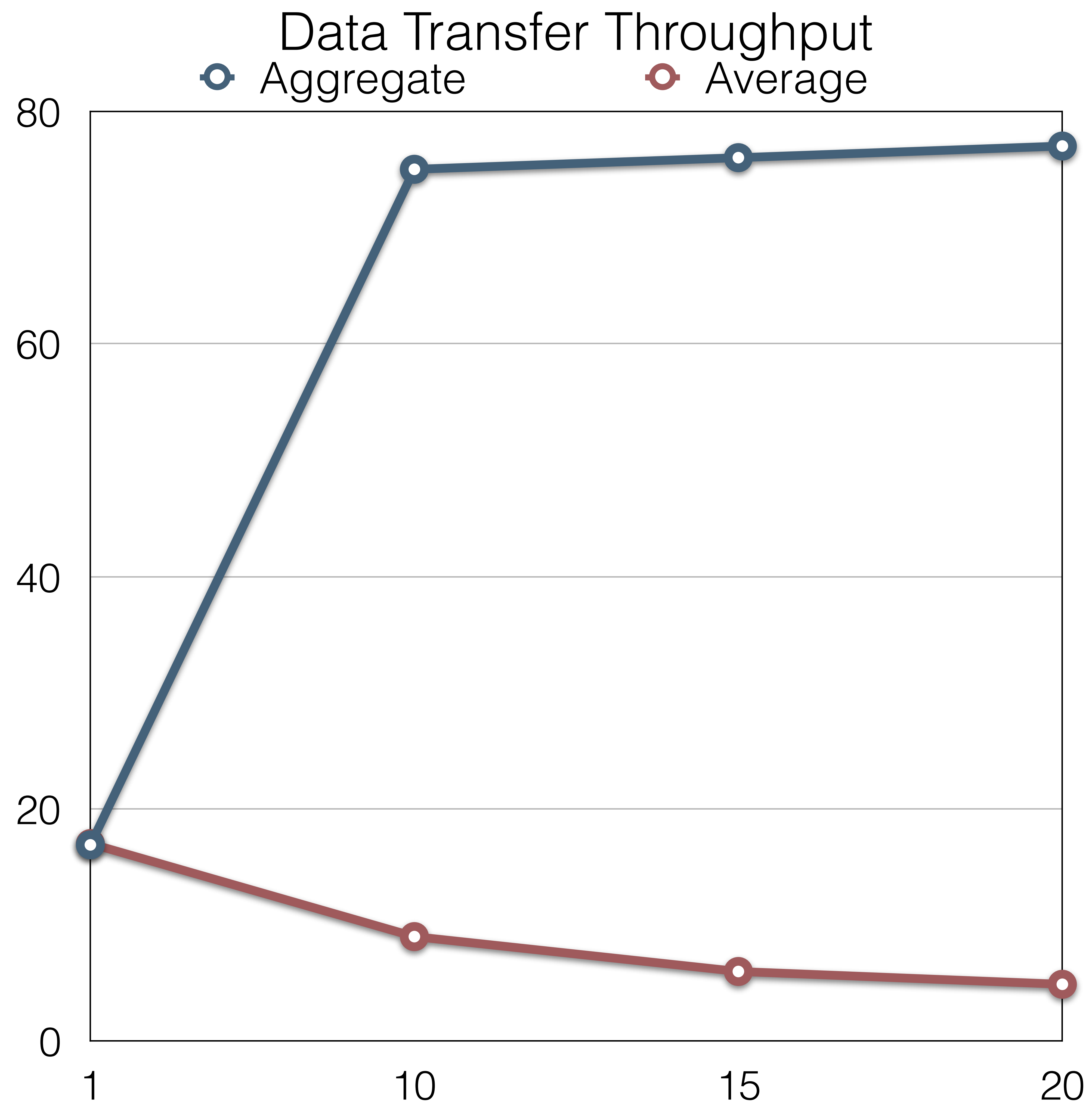
Maximum price $ 0.50

Normally $2.52/Hour

Spot price: $0.50/Hour!

Performance on the faster machine: much better

		Per Thread		Agregate	
Threads	Data Uploaded	Total Clock Time Σ(threads)	Avg throughput per thread	Wall Clock Time	Aggregate Througput
1	18GB	1087	17.0 MB/s	1087	16.89 MB/s
2	16 GB	1288	14.8 MB/s	644	26.34 MB/s
10	15 GB	1623	9 MB/s	194	75 MB/s
15	14 GB	2219	6 MB/s	186	76 MB/s
20	17 GB	3446	4.9 MB/s	224	77 MB/s



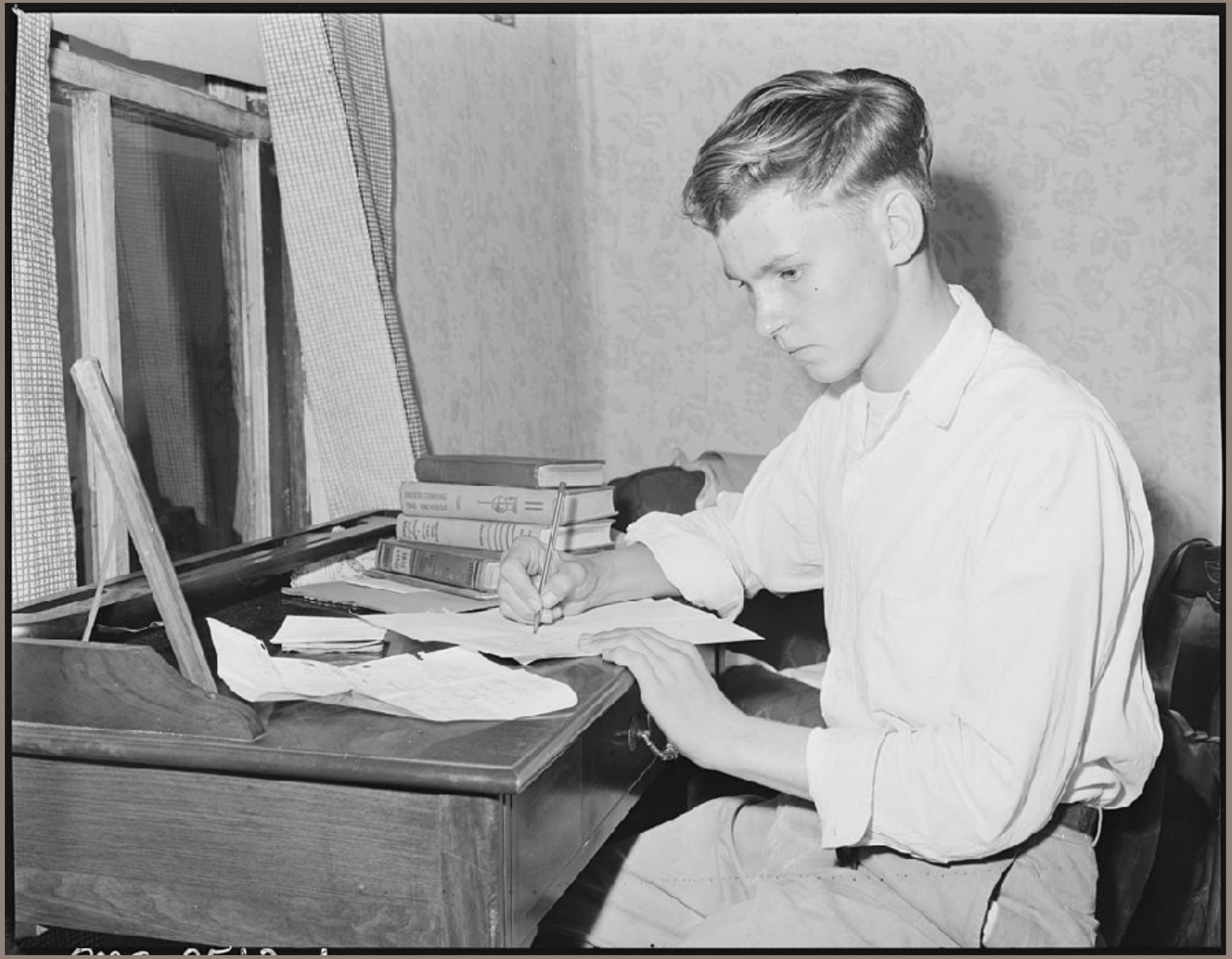
Notice: adding more threads improved performance, but not beyond 75 MB/sec

- New bottleneck: remote server?

Contact Information

Simson L. Garfinkel
sg1224@georgetown.edu
Personal Cell: 202-322-8411

Marck Vaisman
mv559@georgetown.edu
Mobile: (954) 599-1056



http://bit.ly/louis_sergent_homework_1946

A1 — January 20

Technologies you should know

REQUIRED:

git

Text editor

- emacs / VIM / nano / BBEdit

Amazon Web Services — Starting, Monitoring & Shutting Down VMs

- Web interface / Graphical User Interface (GUI)
- Command Line Interface (CLI)

OPTIONAL:

PyCharm

Homework — Getting set up

Git repo: https://bitbucket.org/ANLY502/anly502_2017_spring.git

```
[Dance ~/gits/anly502_2017 15:30:18](master) $ ls -l
total 8
drwxr-xr-x  13 simsong  staff   442 Jan  5 00:10 A0/
drwxr-xr-x  10 simsong  staff   340 Jan  5 15:22 A1/
drwxr-xr-x   4 simsong  staff   136 Jan  2 10:13 A2/
drwxr-xr-x   3 simsong  staff   102 Jan  2 10:14 A3/
drwxr-xr-x   3 simsong  staff   102 Jan  2 10:14 A4/
drwxr-xr-x   3 simsong  staff   102 Jan  2 10:15 A5/
drwxr-xr-x   4 simsong  staff   136 Jan  2 12:19 L01/
drwxr-xr-x   4 simsong  staff   136 Jan  2 12:20 L02/
drwxr-xr-x   2 simsong  staff    68 Jan  2 12:19 L03/
-rw-r--r--   1 simsong  staff  1636 Jan  2 12:21 README.md
drwxr-xr-x  21 simsong  staff   714 Jan  5 00:10 lib/
-rw-r--r--   1 simsong  staff    89 Jan  2 21:16 user.cfg
[Dance ~/gits/anly502_2017 15:30:19](master) $
```

Fork (or clone) the repo.

- Make sure that your personal repo is not world readable

Edit the file user.cfg and insert your information:

```
#
# Enter your user information
#
[USER]
name: Pat Student
email: pat@georgetown.edu
```

Homework — Assignment #1 —

A1/README.md — The assignment in Mark Down

A1/Makefile — The assignment “build system”

```
#
# Parameters
export ASSIGNMENT=A1
export REQUIRED_FILES=q1.txt q2.txt q3.txt q4.txt q5.txt q6.txt
export OPTIONAL_FILES=

#
# Figure out which python we can use
PYTHON3=$(shell which python35 || which python3.5 || which python34 || which python3.4 || echo python3)

check:
	@$(PYTHON3) -c "print('Python3 is operational');"
	@$(PYTHON3) validator.py --check

submit:
	$(PYTHON3) validator.py --zip --check
```

Environment variables get passed to programs

Find Python 3

To make zip file for submission

```
$ make submit
```

Note: Name is *GeorgetownID-assignment.zip*

- *The script names it automatically, and checks your syntax, and perhaps does more!*

Lab

Create an SSH public/private key pair

Create an AWS account

Launch a T1.micro VM

Log into it with -A to proxy your public key

Install Python3 (e.g. “sudo yum install python35”)

Create a BitBucket account

Add your SSH public key

Fork the BitBucket class repo

Check out your fork of the repo on the VM

Modify a file

Commit the file

Push it back to your repo