

Case Study: Amazon AWS

CSE 40822 – Cloud Computing

Prof. Douglas Thain

University of Notre Dame

Caution to the Reader:

Herein are examples of prices consulted in spring 2016, to give a sense of the magnitude of costs. Do your own research before spending your own money!

Several Historical Trends

- Shared Utility Computing
 - 1960s – MULTICS – Concept of a Shared Computing Utility
 - 1970s – IBM Mainframes – rent by the CPU-hour. (Fast/slow switch.)
- Data Center Co-location
 - 1990s-2000s – Rent machines for months/years, keep them close to the network access point and pay a flat rate. Avoid running your own building with utilities!
- Pay as You Go
 - Early 2000s - Submit jobs to a remote service provider where they run on the raw hardware. Sun Cloud (\$1/CPU-hour, Solaris +SGE) IBM Deep Capacity Computing on Demand (50 cents/hour)
- Virtualization
 - 1960s – OS-VM, VM-360 – Used to split mainframes into logical partitions.
 - 1998 – VMWare – First practical implementation on X86, but at significant performance hit.
 - 2003 – Xen paravirtualization provides much perf, but kernel must assist.
 - Late 2000s – Intel and AMD add hardware support for virtualization.

Virtual-* Allows for the Scale of Abstraction to Increase Over Time

- Run one process within certain resource limits.
Op Sys has virtual memory, virtual CPU, and virtual storage (file system).
- Run multiple processes within certain resource limits.
Resource containers (Solaris), virtual servers (Linux), virtual images (Docker)
- Run an entire operating system within certain limits.
Virtual machine technology: VMWare, Xen, KVM, etc.
- Run a set of virtual machines connected via a private network.
Virtual networks (SDNs) provision bandwidth between virtual machines.
- Run a private virtual architecture for every customer.
Automated tools replicate virtual infrastructure as needed.

Amazon AWS

- Grew out of Amazon's need to rapidly provision and configure machines of standard configurations for its own business.
- Early 2000s – Both private and shared data centers began using virtualization to perform “server consolidation”
- 2003 – Internal memo by Chris Pinkham describing an “infrastructure service for the world.”
- 2006 – S3 first deployed in the spring, EC2 in the fall
- 2008 – Elastic Block Store available.
- 2009 – Relational Database Service
- 2012 – DynamoDB
- **Does it turn a profit?**



Amazon Web Services

Compute & Networking

- Direct Connect**
Dedicated Network Connection to AWS
- EC2**
Virtual Servers in the Cloud
- Route 53**
Scalable Domain Name System
- VPC**
Isolated Cloud Resources

Storage & Content Delivery

- CloudFront**
Global Content Delivery Network
- Glacier**
Archive Storage in the Cloud
- S3**
Scalable Storage in the Cloud
- Storage Gateway**
Integrates On-Premises IT Environments with Cloud Storage

Database

- DynamoDB**
Predictable and Scalable NoSQL Data Store
- ElastiCache**
In-Memory Cache
- RDS**
Managed Relational Database Service
- Redshift**
Managed Petabyte-Scale Data Warehouse Service

Deployment & Management

- CloudFormation**
Templated AWS Resource Creation
- CloudTrail**
User Activity and Change Tracking
- CloudWatch**
Resource and Application Monitoring
- Directory Service**
Managed Directories in the Cloud
- Elastic Beanstalk**
AWS Application Container
- IAM**
Secure AWS Access Control
- OpsWorks**
DevOps Application Management Service
- Trusted Advisor**
AWS Cloud Optimization Expert

Analytics

- Data Pipeline**
Orchestration for Data-Driven Workflows
- Elastic MapReduce**
Managed Hadoop Framework
- Kinesis**
Real-time Processing of Streaming Big Data

Mobile Services

- Cognito**
User Identity and App Data Synchronization
- Mobile Analytics**
Understand App Usage Data at Scale
- SNS**
Push Notification Service

App Services

- 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

Applications

- WorkSpaces**
Desktops in the Cloud
- Zocalo**
Secure Enterprise Storage and Sharing Service

Additional Resources

Getting Started

See our documentation to get started and learn more about how to use our services.

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.

Service Health

All services operating normally.

Updated: Oct 26 2014 21:31:00 GMT-0400

[Service Health Dashboard](#)

Set Start Page

Console Home

Terminology

- Instance = One running virtual machine.
- Instance Type = hardware configuration: cores, memory, disk.
- Instance Store Volume = Temporary disk associated with instance.
- Image (AMI) = Stored bits which can be turned into instances.
- Key Pair = Credentials used to access VM from command line.
- Region = Geographic location, price, laws, network locality.
- Availability Zone = Subdivision of region that is fault-independent.

Model	vCPU	CPU Credits / hour	Mem (GiB)	Storage (GB)
t2.micro	1	6	1	EBS Only
t2.small	1	12	2	EBS Only
t2.medium	2	24	4	EBS Only

Model	vCPU	Mem (GiB)	SSD Storage (GB)
c3.large	2	3.75	2 x 16
c3.xlarge	4	7.5	2 x 40
c3.2xlarge	8	15	2 x 80
c3.4xlarge	16	30	2 x 160
c3.8xlarge	32	60	2 x 320

Use Cases

High performance front-end fleets, web-servers, on-demand batch processing, distributed analytics, high performance science and engineering applications, ad serving, batch processing, MMO gaming, video encoding, and distributed analytics.

Model	vCPU	Mem (GiB)	SSD Storage (GB)
m3.medium	1	3.75	1 x 4
m3.large	2	7.5	1 x 32
m3.xlarge	4	15	2 x 40
m3.2xlarge	8	30	2 x 80

Model	vCPU	Mem (GiB)	SSD Storage (GB)
r3.large	2	15.25	1 x 32
r3.xlarge	4	30.5	1 x 80
r3.2xlarge	8	61	1 x 160
r3.4xlarge	16	122	1 x 320
r3.8xlarge	32	244	2 x 320

Use Cases

We recommend memory-optimized instances for high performance databases, distributed memory caches, in-memory analytics, genome assembly and analysis, larger deployments of SAP, Microsoft SharePoint, and other enterprise applications.

EC2 Pricing Model

- Free Usage Tier
- On-Demand Instances
 - Start and stop instances whenever you like, costs are rounded up to the nearest hour. (Worst price)
- Reserved Instances
 - Pay up front for one/three years in advance. (Best price)
 - Unused instances can be sold on a secondary market.
- Spot Instances
 - Specify the price you are willing to pay, and instances get started and stopped without any warning as the market changes. (Kind of like Condor!)

<http://aws.amazon.com/ec2/pricing/>

Free Usage Tier

- 750 hours of EC2 running Linux, RHEL, or SLES t2.micro instance usage
- 750 hours of EC2 running Microsoft Windows Server t2.micro instance usage
- 750 hours of Elastic Load Balancing plus 15 GB data processing
- 30 GB of Amazon Elastic Block Storage in any combination of General Purpose (SSD) or Magnetic, plus 2 million I/Os (with Magnetic) and 1 GB of snapshot storage
- 15 GB of bandwidth out aggregated across all AWS services
- 1 GB of Regional Data Transfer

On-Demand Instance Prices

Linux

RHEL

SLES

Windows

Windows with SQL Standard

Windows with SQL Web

Windows with SQL Enterprise

Region:

US East (N. Virginia)

vCPU

ECU

Memory (GiB)

Instance Storage (GB)

Linux/UNIX Usage

General Purpose - Current Generation

t2.nano	1	Variable	0.5	EBS Only	\$0.0065 per Hour
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	\$0.12 per Hour
m4.xlarge	4	13	16	EBS Only	\$0.239 per Hour
m4.2xlarge	8	26	32	EBS Only	\$0.479 per Hour
m4.4xlarge	16	53.5	64	EBS Only	\$0.958 per Hour
m4.10xlarge	40	124.5	160	EBS Only	\$2.394 per Hour

Reserved Instance Example

t2.medium

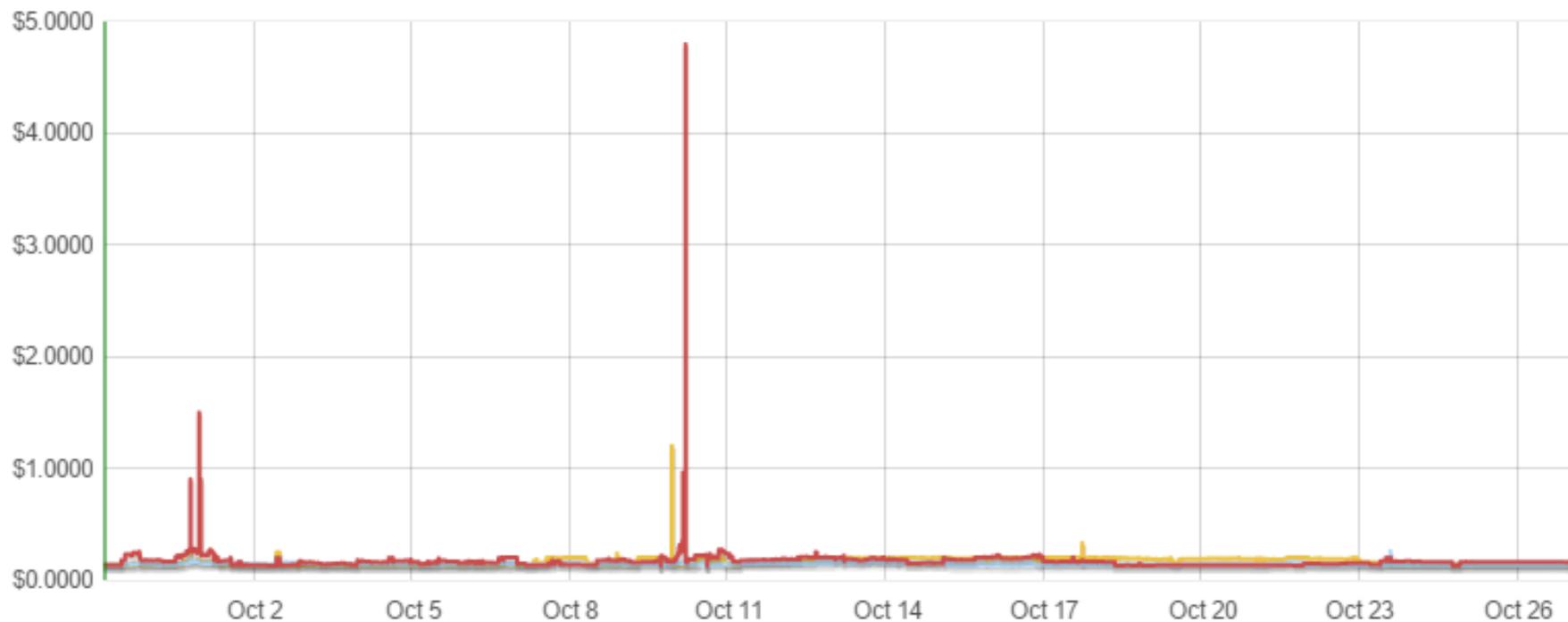
1-YEAR TERM					
Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
No Upfront	\$0	\$26.28	\$0.036	31%	\$0.052 per Hour
Partial Upfront	\$204	\$8.76	\$0.0353	32%	
All Upfront	\$302	\$0	\$0.0345	34%	

3-YEAR TERM					
Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
Partial Upfront	\$436	\$5.84	\$0.0246	53%	\$0.052 per Hour
All Upfront	\$607	\$0	\$0.0231	56%	

Spot Instance Pricing History



Product : **Linux/UNIX** ▾ Instance type: **c3.4xlarge** ▾ Date range : **1 month** ▾ Availability zone: **All zones** ▾



Availability zone	Price
 us-west-2a	\$0.1389
 us-west-2b	\$0.1390
 us-west-2c	\$0.1322

Date September 28, 2014 9:08:53 AM UTC-4

Surprisingly, you can't scale up that large.

Q: How many instances can I run in Amazon EC2?

You are limited to running up to 20 On-Demand instances, purchasing 20 Reserved Instances, and requesting Spot Instances per your [dynamic Spot limit](#) per region. New AWS accounts may start with limits that are lower than the limits described here. Certain instance types are further limited per region as follows:

Instance Type	On-Demand Limit	Reserved Limit	Spot Limit
m4.4xlarge	10	20	Dynamic Spot Limit
m4.10xlarge	5	20	Dynamic Spot Limit
c4.4xlarge	10	20	Dynamic Spot Limit
c4.8xlarge	5	20	Dynamic Spot Limit
cg1.4xlarge	2	20	Dynamic Spot Limit
hi1.4xlarge	2	20	Dynamic Spot Limit
hs1.8xlarge	2	20	Not offered
cr1.8xlarge	2	20	Dynamic Spot Limit
r3.4xlarge	5	20	Dynamic Spot Limit

Simple Storage Service (S3)

- A **bucket** is a container for objects and describes location, logging, accounting, and access control. A bucket can hold any number of **objects**, which are files of up to 5TB. A bucket has a name that must be **globally unique**.
- Fundamental operations corresponding to HTTP actions:
 - `http://bucket.s3.amazonaws.com/object`
 - POST a new object or update an existing object.
 - GET an existing object from a bucket.
 - DELETE an object from the bucket
 - LIST keys present in a bucket, with a filter.
- A bucket has a **flat directory structure** (despite the appearance given by the interactive web interface.)

Easily Integrated into Web Applications

```
<form action="http://examplebucket.s3.amazonaws.com/" method="post" enctype="multipart/form-data">  
  
<input type="input" name="key" value="user/user1/" />  
  
<input type="hidden" name="acl" value="public-read" />  
<input type="hidden" name="success_action_redirect"  
    value="http://examplebucket.s3.amazonaws.com/successful_upload.html" />  
...  
<input type="text" name="X-Amz-Credential"  
    value="AKIAIOSFODNN7EXAMPLE/20130806/us-east-1/s3/aws4_request" />  
...  
<input type="submit" name="submit" value="Upload to Amazon S3" /> </form>
```

<http://docs.aws.amazon.com/AmazonS3/latest/API/sigv4-post-example.html>

Bucket Properties

- Versioning – If enabled, POST/DELETE result in the creation of new versions without destroying the old.
- Lifecycle – Delete or archive objects in a bucket a certain time after creation or last access or number of versions.
- Access Policy – Control **when and where** objects can be accessed.
- Access Control – Control who **may** access objects in this bucket.
- Logging – Keep track of how objects are accessed.
- Notification – Be notified when failures occur.

S3 Weak Consistency Model

Direct quote from the Amazon developer API:

“Updates to a single key are **atomic**....”

“Amazon S3 achieves high availability by replicating data across multiple servers within Amazon's data centers. If a PUT request is successful, your data is safely stored. However, information about the changes must replicate across Amazon S3, which can take some time, and so you might observe the following behaviors:

- A process writes a new object to Amazon S3 and immediately attempts to read it. Until the change is fully propagated, Amazon S3 might report "key does not exist."
- A process writes a new object to Amazon S3 and immediately lists keys within its bucket. Until the change is fully propagated, the object might not appear in the list.
- A process replaces an existing object and immediately attempts to read it. Until the change is fully propagated, Amazon S3 might return the prior data.
- A process deletes an existing object and immediately attempts to read it. Until the deletion is fully propagated, Amazon S3 might return the deleted data.”

Storage Pricing

Region:

US East (N. Virginia)



	Standard Storage	Standard - Infrequent Access Storage †	Glacier Storage
First 1 TB / month	\$0.0300 per GB	\$0.0125 per GB	\$0.007 per GB
Next 49 TB / month	\$0.0295 per GB	\$0.0125 per GB	\$0.007 per GB
Next 450 TB / month	\$0.0290 per GB	\$0.0125 per GB	\$0.007 per GB
Next 500 TB / month	\$0.0285 per GB	\$0.0125 per GB	\$0.007 per GB
Next 4000 TB / month	\$0.0280 per GB	\$0.0125 per GB	\$0.007 per GB
Over 5000 TB / month	\$0.0275 per GB	\$0.0125 per GB	\$0.007 per GB

Request Pricing

Region: US East (N. Virginia) ▾

Pricing

For Requests Not Otherwise Specified Below

PUT, COPY, POST, or LIST Requests	\$0.005 per 1,000 requests
GET and all other Requests	\$0.004 per 10,000 requests
Delete Requests	Free †

For Standard – Infrequent Access Requests

PUT, COPY, or POST Requests	\$0.01 per 1,000 requests
GET and all other Requests	\$0.01 per 10,000 requests
Lifecycle Transition Requests into Standard – Infrequent Access	\$0.01 per 1,000 requests
Data Retrievals	\$0.01 per GB

For Glacier Requests

Glacier Archive and Restore Requests	\$0.05 per 1,000 requests
Glacier Data Restores	Free †

Always read the fine print....

† No charge for delete requests of Standard objects. For objects that are archived to Glacier, there is a pro-rated charge of \$0.021 per gigabyte for objects deleted prior to 90 days. [Learn more](#). Objects that are in Standard – Infrequent Access have a minimum 30 days of storage, and objects that are deleted, overwritten, or transitioned to a different storage class before 30 days incur a pro-rated charge equal to the storage charge for the remaining days. [Learn more](#).

‡ Glacier is designed with the expectation that restores are infrequent and unusual, and data will be stored for extended periods of time. You can restore up to 5% of your average monthly Glacier storage (pro-rated daily) for free each month. If you choose to restore more than this amount of data in a month, you are charged a restore fee starting at \$0.01 per gigabyte. [Learn more](#).

Data Transfer IN To Amazon S3

All data transfer in

\$0.000 per GB

Data Transfer OUT From Amazon S3 To

Amazon EC2 in the Northern Virginia Region

\$0.000 per GB

Another AWS Region

\$0.020 per GB

Amazon CloudFront

\$0.000 per GB

Data Transfer OUT From Amazon S3 To Internet

First 1 GB / month

\$0.000 per GB

Up to 10 TB / month

\$0.090 per GB

Next 40 TB / month

\$0.085 per GB

Next 100 TB / month

\$0.070 per GB

Next 350 TB / month

\$0.050 per GB

Next 524 TB / month

[Contact Us](#)

Next 4 PB / month

[Contact Us](#)

Greater than 5 PB / month

[Contact Us](#)

Elastic Block Store

- An EBS volume is a **virtual disk** of a fixed size with a block read/write interface. It can be **mounted** as a filesystem on a running EC2 instance where it can be **updated incrementally**. Unlike an instance store, an EBS volume is **persistent**.
- (Compare to an S3 object, which is essentially a file that must be accessed in its entirety.)
- Fundamental operations:
 - CREATE a new volume (1GB-1TB)
 - COPY a volume from an existing EBS volume or S3 object.
 - MOUNT on one instance at a time.
 - SNAPSHOT current state to an S3 object.

Amazon EBS Volume Types

Volume Type	EBS General Purpose (SSD)	EBS Provisioned IOPS (SSD)	EBS Magnetic
Use Cases	<ul style="list-style-type: none"> Boot volumes Small to Med DBs Dev and Test 	<ul style="list-style-type: none"> I/O intensive Relational DBs NoSQL DBs 	<ul style="list-style-type: none"> Infrequent Data Access
Storage Media	SSD-backed	SSD-backed	Magnetic disk-backed
Max Volume Size	1TB	1TB	1TB
Max IOPS/volume	3,000 (burst)	4,000	40 - 200
Max throughput/volume	128MBps	128MBps	40 - 90MBps
Max IOPS/instance	48,000	48,000	48,000
Max throughput/instance	800MBps	800MBps	800MBps
API Name	gp2	io1	standard
Price*	\$.10/GB - Month	\$.125/GB - Month \$.065/provisioned IOPS	\$.05/GB - Month \$.05/million I/O

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 is approx. 3x more expensive by volume and 10x more expensive by IOPS than S3.

Use Glacier for Cold Data

- Glacier is structured like S3: a **vault** is a container for an arbitrary number of **archives**. Policies, accounting, and access control are associated with vaults, while an archive is a single object.
- However:
 - All operations are asynchronous and notified via SNS.
 - Vault listings are updated once per day.
 - Archive downloads may take up to four hours.
 - Only 5% of total data can be accessed in a given month.
- Pricing:
 - Storage: \$0.01 per GB-month
 - Operations: \$0.05 per 1000 requests
 - Data Transfer: Like S3, free within AWS.
- S3 Policies can be set up to automatically move data into Glacier.

Durability

- Amazon claims about S3:
 - Amazon S3 is designed to sustain the concurrent loss of data in two facilities, e.g. 3+ copies across multiple available domains.
 - 99.999999999% durability of objects over a given year.
- Amazon claims about EBS:
 - Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component.
 - Volumes <20GB modified data since last snapshot have an annual failure rate of 0.1% - 0.5%, resulting in complete loss of the volume.
 - Commodity hard disks have an AFR of about 4%.
- Amazon claims about Glacier is the same as S3:
 - Amazon S3 is designed to sustain the concurrent loss of data in two facilities, e.g. 3+ copies across multiple available domains PLUS periodic internal integrity checks.
 - 99.999999999% durability of objects over a given year.
- **Beware of oversimplified arguments about low-probability events!**

Architecture Center

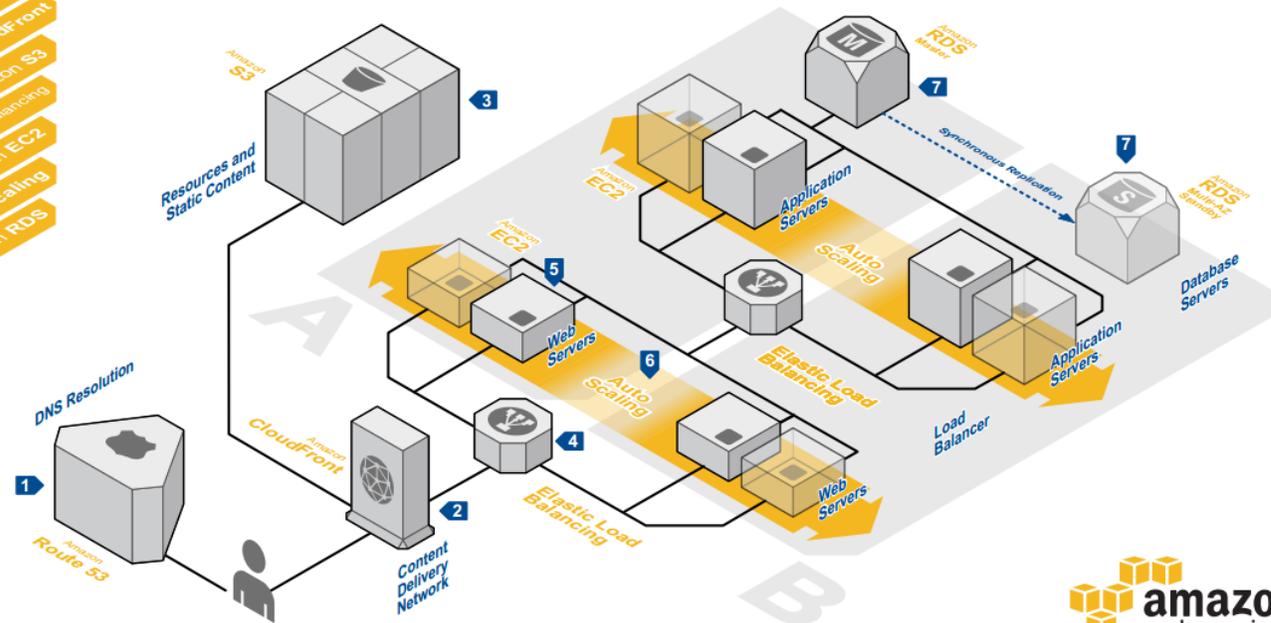
- Ideas for constructing large scale infrastructures using AWS:

<http://aws.amazon.com/architecture/>

AWS Reference Architectures
Amazon Route 53
Amazon CloudFront
Amazon S3
Elastic Load Balancing
Amazon EC2
Auto scaling
Amazon RDS

WEB APPLICATION HOSTING

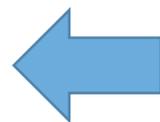
Highly available and scalable web hosting can be complex and expensive. Dense peak periods and wild swings in traffic patterns result in low utilization of expensive hardware. Amazon Web Services provides the reliable, scalable, secure, and high-performance infrastructure required for web applications while enabling an elastic, scale-out and scale-down infrastructure to match IT costs in real time as customer traffic fluctuates.



Command Line Setup

- Go to your profile menu (your name) in the upper right hand corner, select “Security Credentials” and “Continue to Security Credentials”
- Select “Access Keys”
- Select “New Access Key” and save the generated keys somewhere.
- Edit `~/.aws/config` and set it up like this:

```
[default]
output = json
region = us-west-2
aws_access_key = XXXXXX
aws_secret_access_key = YYYYYYYYYYYY
```



Note the syntax here is different from how it was given in the web console!

```
AWSAccessKey=XXXXXX
AWSSecretAccessKey=YYYYYYYYYY
```

- Now test it: **`aws ec2-describe-instances`**

S3 Command Line Examples

```
aws s3 mb s3://bucket
...   cp localfile s3://bucket/key
      mv s3://bucket/key s3://bucket/newname
      ls s3://bucket
      rm s3://bucket/key
      rb s3://bucket
```

```
aws s3 help
aws s3 ls help
```

EC2 Command Line Examples

```
aws ec2 describe-instances
```

```
run-instances --image-id ami-xxxxx -- count 1
```

```
    --instance-type t1.micro --key-name keyfile
```

```
stop-instances --instance-id i-xxxxxx
```

```
aws ec2 help
```

```
aws ec2 start-instances help
```

Warmup: Get Started with Amazon

- Skim through the AWS documentation.
- Sign up for AWS at <http://aws.amazon.com>
- (Skip the IAM management for now)
- Apply the service credit you received by email.
- Create and download a Key-Pair, save it in your home directory.
- Create a VM via the AWS Console
- Connect to your newly-created VM like this:
 - `ssh -i my-aws-keypair.pem ec2-user@ip-address-of-vm`
- Create a bucket in S3 and upload/download some files.

Demo Time

<http://aws.amazon.com>