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 fall 2018, 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
- 2014 – Lambda ("Serverless")
- 2016 (?) – Elastic Container Service
- **Does it turn a profit?**
Amazon Web Services

Compute & Networking
- Direct Connect
- EC2
- Route 53
- VPC

Storage & Content Delivery
- CloudFront
- Glacier
- S3
- Storage Gateway

Database
- DynamoDB
- ElastiCache
- RDS
- Redshift

Deployment & Management
- CloudFormation
- CloudTrail
- CloudWatch
- Directory Service
- Elastic Beanstalk
- IAM
- OpsWorks
- Trusted Advisor

Analytics
- Data Pipeline
- Elastic MapReduce
- Kinesis

Mobile Services
- Cognito
- Mobile Analytics
- SNS

App Services
- AppStream
- CloudSearch
- Elastic Transcoder
- SES
- SWF
- WorkSpaces

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-0500

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.
<table>
<thead>
<tr>
<th>Model</th>
<th>vCPU</th>
<th>CPU Credits / hour</th>
<th>Mem (GiB)</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.micro</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>EBS Only</td>
</tr>
<tr>
<td>t2.small</td>
<td>1</td>
<td>12</td>
<td>2</td>
<td>EBS Only</td>
</tr>
<tr>
<td>t2.medium</td>
<td>2</td>
<td>24</td>
<td>4</td>
<td>EBS Only</td>
</tr>
<tr>
<td>m3.medium</td>
<td>1</td>
<td>3.75</td>
<td>1</td>
<td>1 x 4</td>
</tr>
<tr>
<td>m3.large</td>
<td>2</td>
<td>7.5</td>
<td>1</td>
<td>1 x 32</td>
</tr>
<tr>
<td>m3.xlarge</td>
<td>4</td>
<td>15</td>
<td>2</td>
<td>2 x 40</td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>8</td>
<td>30</td>
<td>2</td>
<td>2 x 80</td>
</tr>
<tr>
<td>r3.large</td>
<td>2</td>
<td>15.25</td>
<td>1</td>
<td>1 x 32</td>
</tr>
<tr>
<td>r3.xlarge</td>
<td>4</td>
<td>30.5</td>
<td>1</td>
<td>1 x 80</td>
</tr>
<tr>
<td>r3.2xlarge</td>
<td>8</td>
<td>61</td>
<td>1</td>
<td>1 x 160</td>
</tr>
<tr>
<td>r3.4xlarge</td>
<td>16</td>
<td>122</td>
<td>1</td>
<td>1 x 320</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>32</td>
<td>244</td>
<td>2</td>
<td>2 x 320</td>
</tr>
</tbody>
</table>

**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.

**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 marked 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 Instances

### General Purpose - Current Generation

<table>
<thead>
<tr>
<th>Instance</th>
<th>vCPU</th>
<th>ECU</th>
<th>Memory (GiB)</th>
<th>Instance Storage (GB)</th>
<th>Linux/UNIX Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3.nano</td>
<td>2</td>
<td>Variable</td>
<td>0.5 GiB</td>
<td>EBS Only</td>
<td>$0.0052 per Hour</td>
</tr>
<tr>
<td>t3.micro</td>
<td>2</td>
<td>Variable</td>
<td>1 GiB</td>
<td>EBS Only</td>
<td>$0.0104 per Hour</td>
</tr>
<tr>
<td>t3.small</td>
<td>2</td>
<td>Variable</td>
<td>2 GiB</td>
<td>EBS Only</td>
<td>$0.0208 per Hour</td>
</tr>
<tr>
<td>t3.medium</td>
<td>2</td>
<td>Variable</td>
<td>4 GiB</td>
<td>EBS Only</td>
<td>$0.0416 per Hour</td>
</tr>
<tr>
<td>t3.large</td>
<td>2</td>
<td>Variable</td>
<td>8 GiB</td>
<td>EBS Only</td>
<td>$0.0832 per Hour</td>
</tr>
<tr>
<td>t3.xlarge</td>
<td>4</td>
<td>Variable</td>
<td>16 GiB</td>
<td>EBS Only</td>
<td>$0.1664 per Hour</td>
</tr>
<tr>
<td>t3.2xlarge</td>
<td>8</td>
<td>Variable</td>
<td>32 GiB</td>
<td>EBS Only</td>
<td>$0.3328 per Hour</td>
</tr>
</tbody>
</table>
## t3.large

### STANDARD 1-YEAR TERM

<table>
<thead>
<tr>
<th>Payment Option</th>
<th>Upfront</th>
<th>Monthly*</th>
<th>Effective Hourly**</th>
<th>Savings over On-Demand</th>
<th>On-Demand Hourly</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Upfront</td>
<td>$0.00</td>
<td>$38.11</td>
<td>$0.052</td>
<td>37%</td>
<td></td>
</tr>
<tr>
<td>Partial Upfront</td>
<td>$218.00</td>
<td>$18.10</td>
<td>$0.05</td>
<td>40%</td>
<td>$0.0832</td>
</tr>
<tr>
<td>All Upfront</td>
<td>$426.00</td>
<td>$0.00</td>
<td>$0.049</td>
<td>42%</td>
<td></td>
</tr>
</tbody>
</table>

### CONVERTIBLE 1-YEAR TERM

<table>
<thead>
<tr>
<th>Payment Option</th>
<th>Upfront</th>
<th>Monthly*</th>
<th>Effective Hourly**</th>
<th>Savings over On-Demand</th>
<th>On-Demand Hourly</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Upfront</td>
<td>$0.00</td>
<td>$43.80</td>
<td>$0.06</td>
<td>28%</td>
<td></td>
</tr>
<tr>
<td>Partial Upfront</td>
<td>$250.00</td>
<td>$20.88</td>
<td>$0.057</td>
<td>31%</td>
<td>$0.0832</td>
</tr>
<tr>
<td>All Upfront</td>
<td>$490.00</td>
<td>$0.00</td>
<td>$0.056</td>
<td>33%</td>
<td></td>
</tr>
</tbody>
</table>

### STANDARD 3-YEAR TERM

<table>
<thead>
<tr>
<th>Payment Option</th>
<th>Upfront</th>
<th>Monthly*</th>
<th>Effective Hourly**</th>
<th>Savings over On-Demand</th>
<th>On-Demand Hourly</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Upfront</td>
<td>$0.00</td>
<td>$26.28</td>
<td>$0.036</td>
<td>57%</td>
<td></td>
</tr>
<tr>
<td>Partial Upfront</td>
<td>$438.00</td>
<td>$12.19</td>
<td>$0.033</td>
<td>60%</td>
<td>$0.0832</td>
</tr>
<tr>
<td>All Upfront</td>
<td>$823.00</td>
<td>$0.00</td>
<td>$0.031</td>
<td>62%</td>
<td></td>
</tr>
</tbody>
</table>
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 5 Spot Instances 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:

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>On-Demand Limit</th>
<th>Reserved Limit</th>
<th>Spot Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>cg1.4xlarge</td>
<td>2</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>hi1.4xlarge</td>
<td>2</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>hs1.8xlarge</td>
<td>2</td>
<td>20</td>
<td>Not offered</td>
</tr>
<tr>
<td>cr1.8xlarge</td>
<td>2</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>g2.2xlarge</td>
<td>5</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>
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](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 (varies by region)

<table>
<thead>
<tr>
<th>Region: US East (Ohio)</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Pricing</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S3 Standard Storage</strong></td>
<td></td>
</tr>
<tr>
<td>First 50 TB / Month</td>
<td>$0.023 per GB</td>
</tr>
<tr>
<td>Next 450 TB / Month</td>
<td>$0.022 per GB</td>
</tr>
<tr>
<td>Over 500 TB / Month</td>
<td>$0.021 per GB</td>
</tr>
<tr>
<td><strong>S3 Standard-Infrequent Access (S3 Standard-IA) Storage</strong></td>
<td></td>
</tr>
<tr>
<td>All storage</td>
<td>$0.0125 per GB</td>
</tr>
<tr>
<td><strong>S3 One Zone-Infrequent Access (S3 One Zone-IA) Storage</strong></td>
<td></td>
</tr>
<tr>
<td>All storage</td>
<td>$0.01 per GB</td>
</tr>
<tr>
<td><strong>Amazon Glacier Storage</strong></td>
<td></td>
</tr>
<tr>
<td>All storage</td>
<td>$0.004 per GB</td>
</tr>
</tbody>
</table>
### Request pricing (varies by region)

For requests not otherwise specified below

<table>
<thead>
<tr>
<th>Region: US East (Ohio)</th>
<th>Pricing</th>
</tr>
</thead>
</table>

| Data Returned by S3 Select | $0.0007 per GB |
| Data Scanned by S3 Select  | $0.002 per GB  |
| PUT, COPY, POST, or LIST Requests | $0.005 per 1,000 requests |
| GET, SELECT and all other Requests | $0.0004 per 1,000 requests |
| Lifecycle Transition Requests into Standard – Infrequent Access or One Zone - Infrequent Access | $0.01 per 1,000 requests |

DELETE requests are free. †

Amazon S3 request costs are based on the request type, and are charged on the quantity of requests or the volume of data retrieved as listed in the table above.
Data Transfer Pricing

The pricing below is based on data transferred "in" to and "out" of Amazon S3 (over the public Internet). AWS Direct Connect pricing can be found here. Transfers between S3 buckets or from S3 to any service(s) within the same region are free.

<table>
<thead>
<tr>
<th>Region: US East (Ohio)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Data Transfer IN To Amazon S3 From Internet</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>All data transfer in</td>
<td>$0.00 per GB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Transfer OUT From Amazon S3 To Internet</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 1 GB / Month</td>
<td>$0.00 per GB</td>
</tr>
<tr>
<td>Next 9.999 TB / Month</td>
<td>$0.09 per GB</td>
</tr>
<tr>
<td>Next 40 TB / Month</td>
<td>$0.085 per GB</td>
</tr>
<tr>
<td>Next 100 TB / Month</td>
<td>$0.07 per GB</td>
</tr>
<tr>
<td>Greater than 150 TB / Month</td>
<td>$0.05 per GB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Transfer OUT From Amazon S3 To</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudFront</td>
<td>$0.00 per GB</td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td>$0.01 per GB</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>$0.02 per GB</td>
</tr>
</tbody>
</table>
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 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.
  • 99.9999999999% durability of objects over a given year.

• Beware of oversimplified arguments about low-probability events!
Amazon Elastic File Services (EFS)

• EFS is a standalone file service designed to shared among VMs.
  • File System Instance – Files, directories, storage allocation, multiple AZs
  • Mount Target – DNS name, IP address, NFS target, single AZ:

        file-system.id.efs.aws-region.amazonaws.com

• Inside of VM, use normal NFS connection to the mount target:

        mount -t nfs -o rsize=1M,wsize=1M,...
        file-system-dns-name /mnt/data

https://docs.aws.amazon.com/efs
Amazon ElasticFile Services (EFS)

- Data Consistency Model
  - "close to open" consistency semantics.
  - In (normal) asynchronous, sequential I/O mode:
    - Data is durable after an fsync or a close.
    - Data is visible to other processes that open after you close.
  - In (explicit) synchronous I/O mode (O_DIRECT)
    - Non-appending writes are immediately visible.
    - (This implies that appending writes are not. Why?)
    - Hint: Size of file is a property of metadata.
Everything in AWS is Carefully Limited!

- 50KB/s per GB of data in free throughput
- 250MB/s Max EFS throughput per EC2 Instance
- 128 user IDs can have open files
- 32K open files on a single instance
- 4080B max symbolic link length
- 1000 maximum directory depth
- 47.9TiB max bytes per file
- 87 max locks per file (?)
- 7000 file operations per second

https://docs.aws.amazon.com/efs/latest/ug/limits.html
## Amazon EFS Pricing

<table>
<thead>
<tr>
<th>Region</th>
<th>Storage (GB-Month)</th>
<th>Provisioned Throughput (MB/s-Month)</th>
<th>EFS File Sync (Per-GB into EFS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>$0.30</td>
<td>$6.00</td>
<td>$0.01</td>
</tr>
<tr>
<td>US East (Ohio)</td>
<td>$0.30</td>
<td>$6.00</td>
<td>$0.01</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>$0.33</td>
<td>$6.60</td>
<td>$0.01</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>$0.30</td>
<td>$6.00</td>
<td>$0.01</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>$0.33</td>
<td>$6.60</td>
<td>$0.01</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>$0.36</td>
<td>$7.20</td>
<td>$0.01</td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>$0.36</td>
<td>$7.20</td>
<td>$0.01</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>$0.36</td>
<td>$7.20</td>
<td>$0.01</td>
</tr>
<tr>
<td>EU (Frankfurt)</td>
<td>$0.36</td>
<td>$7.20</td>
<td>$0.01</td>
</tr>
<tr>
<td>EU (Ireland)</td>
<td>$0.33</td>
<td>$6.60</td>
<td>$0.01</td>
</tr>
</tbody>
</table>

Within your first 12 months on AWS, you can use up to 5 GB/month for free.
<table>
<thead>
<tr>
<th></th>
<th>EBS</th>
<th>EFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>&quot;lowest&quot; (local hardware)</td>
<td>&quot;low&quot; (network)</td>
</tr>
<tr>
<td>Max Throughput</td>
<td>2 GB/s</td>
<td>10 GB/s</td>
</tr>
<tr>
<td>Clients (VMs)</td>
<td>Single</td>
<td>O(1000)</td>
</tr>
<tr>
<td>Storage Cost Per Month</td>
<td>$0.05 / GB (Magnetic Disk)</td>
<td>$0.30 / GB</td>
</tr>
<tr>
<td>Access Cost Per Month</td>
<td>$0.05 / 1M IOPS (Magnetic Disk)</td>
<td>$6.00 / MB/s Xput</td>
</tr>
</tbody>
</table>
Architecture Center

• Ideas for constructing large scale infrastructures using AWS:
  http://aws.amazon.com/architecture/
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 = YYYYYYYYYYY
```

Note the syntax here is different from how it was given in the web console!
AWSAccessKey=XXXXXX
AWSSecretAccessKey=YYYYYYYYY

• 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