Amazon Web Services Student Tutorial

David Palma
Joseph Snow

CSC 532:
Advanced Software Engineering
Louisiana Tech University

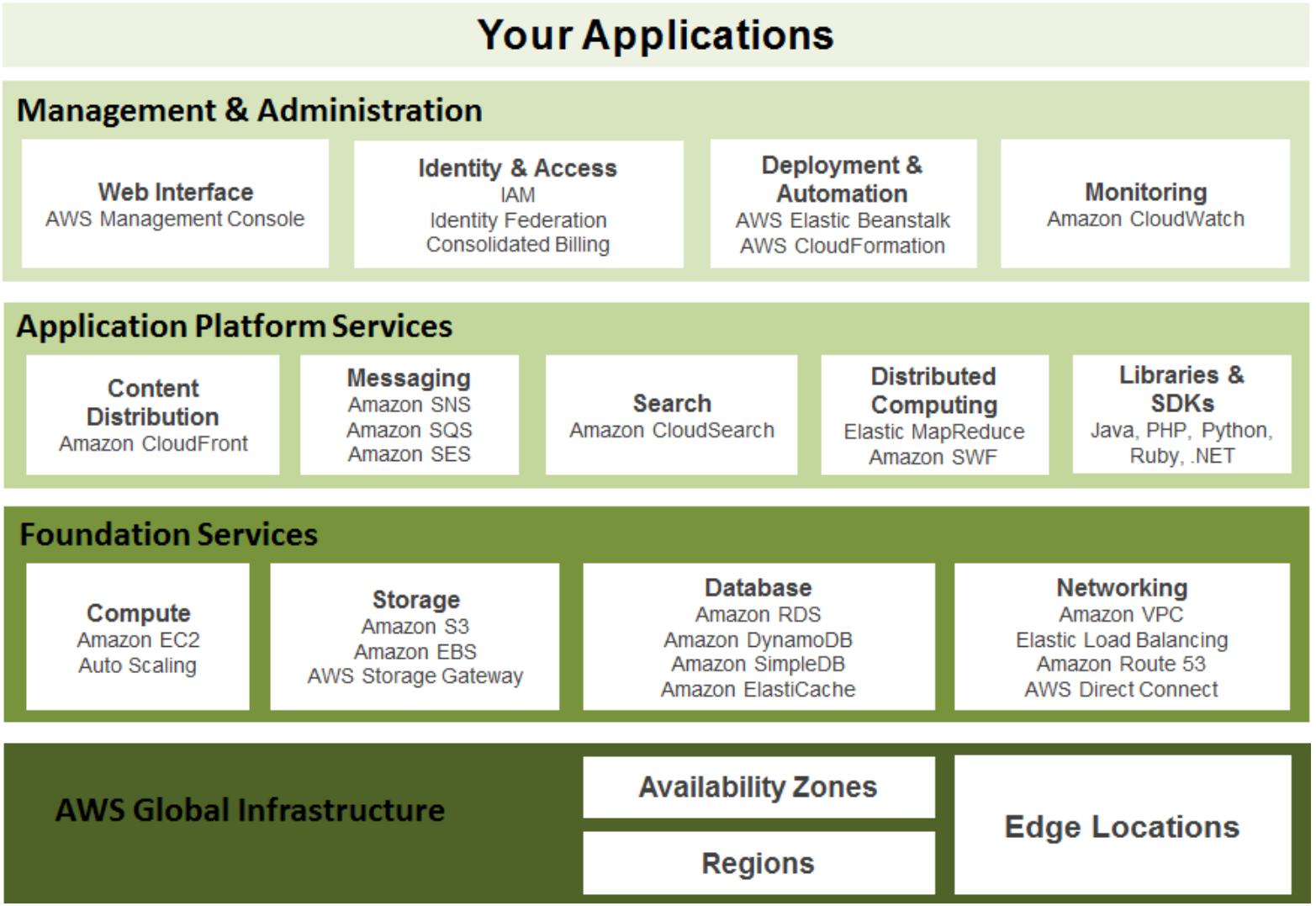
October 4, 2012

Outline

Amazon Web Services

- Amazon Web Services (AWS) provide many distinct services.
- All of AWS are billed pay-as-you-go.
  - pay-as-you-go is commonly phrased as pay-for-what-you-use
- Get an AWS account to develop in the **cloud**.
- Signing up for AWS signs you up all AWS.
- AWS can be managed in a few different ways:
  - AWS command line tools
  - AWS SDKs (varied usage)
  - AWS management console (doesn't do everything)
- Docs: AWS Documentation

Amazon Web Services



Amazon Machine Images (AMIs)

- Essentially a virtual machine image or snapshot
- ”An Amazon Machine Image (AMI) is a special type of pre-configured operating system and virtual application software which is used to create a virtual machine within the Amazon Elastic Compute Cloud (EC2). It serves as the basic unit of deployment for services delivered using EC2.”
- AWS supports the following virtual image types for import/export: VMware ESX **VMDK** images, VMware ESX **OVA** (export only), Citrix **Xen VHD** images and Microsoft **Hyper-V VHD** images. Not free in practice!
- Refs: Amazon Machine Images List
- Refs: VM Import/Export, Import/Export FAQs

Amazon Free Tier

- Amazon Web Services are generally not free; however, AWS does provide a Free Usage Tier for marketing purposes which expires after 12 months from sign up.
- The services available in the Free Usage Tier **also have restrictions/quotas on usage**.
- The majority of the quotas are monthly.
- Other restrictions may apply aside from quotas (e.g. only micro instances are available in the Free Tier)
- Be aware that when you sign up for the Free Tier, you are signing up for all of AWS.
- Docs: Free Usage Tier
- Useful: AWS Simply Monthly Calculator

Budget: $0

- Your Free Tier usage quotas are used up first for billing (you will not be charged for this usage)
- However, once your quotas are exceeded, **you will be billed at normal rates for any further usage**
- Also, any service which you use that is not included in the Free Usage Tier are also billed at normal rates. There is no warning about non-free usage.
- AWS does not provide any way to discontinue services once your free quotas are used up, it is your responsibility to keep track of your usage.
- Keep in mind, if you get charged due to negligence or ”accidents”, you still owe Amazon for the resources you used.

Elastic Compute Cloud (EC2)

- EC2 is the basic fundamental block around which the AWS are structured.
- EC2 provides remote operations of virtual machines (VM) on Amazon's infrastructure.
- A single VM is called an **instance**, and there are different instance types that differ in their available resources.
- A **micro** instance is the only EC2 instance type that is free and is also the most underpowered instance (613MB memory).
- The EC2 micro instance type is the least reliably provisioned; when demand on Amazon's infrastructure is high, the micro instance gets the lowest priority.
- Docs: Instance Types, Micro Instances

Elastic Compute Cloud (EC2)

- Renting an instance on EC2 provides the same functionality as renting a server hosted and maintained by someone else; however, instead of renting **hardware**, you are renting an allocation of **computational resources**
- If EC2 is the basic fundamental block of AWS, then the other services (ex: ELB, EBS, S3, elastic IP, auto-scaling) are the building blocks of AWS.
- An EC2 instance is not scalable by itself.
- If your instance is active, then it consumes as much of its allotted resources as it can.
- To achieve a scalable cloud application using AWS, you must use more services than just EC2.
- Docs: Elastic Compute Cloud

Data Transfer Limits

- Network data transfer is the fundamental resource shared across the Internet.
- More commonly, the term **bandwidth** is used to describe data transfer across a network.
- Several AWS receive data through the Internet or send data across the Internet.
- AWS bills separately for data received (**incoming**) through the Internet and for data sent (**outgoing**) across the Internet.
- Useful: EC2 Pricing (scroll down for Data Transfer)

Data Transfer Limits

- For the services in the Free Usage Tier, all data **incoming** through the Internet is FREE. Huzzah!
- For the services in the Free Usage Tier, all data **outgoing** through the Internet is free for the first **15GB of total outgoing bandwidth**.
- This limit is not applied to each service individually.
- ”There is no data transfer charge between Amazon EC2 and other AWS services within the same region.”
- Aside: AWS regions are related to where (geographically) data is hosted.
- In a nutshell, the only data transfer you pay for is what your application sends out to the Internet.

Data Transfer Usage Case

- Suppose that your application stores 8GB of data on S3.
- The bandwidth to send the data from your application to S3 is free (intraservice bandwidth is free)
- If each of your 4 team members independently download the 8GB of data from S3, then you will have 32GB of outgoing bandwidth.
- The Free Usage Tier only covers 15GB total for all services.
- You would be billed for 17GB outgoing data transfer, and any further outgoing data transfer until the next month.

Elastic Block Store (EBS)

- Block-level storage volumes for use with EC2 instances
- Use them like any block device (like a hard drive).
- EBS volumes are network-attached, and persist independently from the life of an instance
- Once mounted, the instance can interact with the volume as it would with a local drive, formatting it with a file system or installing applications on it directly.
- Storage volume sizes range from 1GB-1TB
- A volume can only be attached to one instance at a time, but many volumes can be attached to a single instance
- Suitable for apps that require a database, file system, or access to raw block level storage.

EBS Snapshots

- EBS provides the ability to back up point-in-time snapshots of your data to S3.
- Snapshots are incremental backups: only the blocks on the device that have changed since your last snapshot will be saved
- **For Example**: If you have a device with 100GBs of data, but only 5GBs of data has changed since your last snapshot, only the 5 additional GBs of snapshot data will be stored back to S3
- Docs: Elastic Block Store
- Free: 30GB storage, 2 million I/Os and 1GB snapshot storage
- Unfortunately, this isn't enough to facilitate AMI import/export despite being a common use case

Simple Storage Service (S3)

- S3 is a minimalistic data hosting service.
- Storing data and downloading data in S3 are billed separately.
- S3 only hosts dumb data (does not change unless you change it)
- S3 is appropriate for:
  - Hosting static web pages (no server-side processing)
  - Hosting large files (e.g. custom AMIs)
- Although S3 is commonly used to store custom AMIs, the Free Usage Tier does not provide enough S3 storage and bandwidth to do so freely.
- Docs: Simple Storage Service
- Free: 5GB storage, 15GB of data outgoing per month

SimpleDB

- SimpleDB is an automatically indexed, non-relational (schema-less) data store.
- Consists of many smaller datasets (limit: 10GB/DB) vs one large DB
- Datasets are organized into query domains
- Domains are collections of items that are described by attribute-value pairs
- Queries run across all data stored in a particular domain
- SimpleDB is not accessible from the management console, must use APIs/SDKs to access
- Docs: SimpleDB
- Free: 25 SimpleDB Machine Hours, 1GB storage per month (most apps should be able to operate perpetually within the free tier limits)

SimpleDB: Intended Usage

- SimpleDB is **not** designed for storing raw data.
- SimpleDB takes your data as input and expands it to create multiple indices, enabling you to quickly query that data
- SimpleDB **is** designed for use with other AWS (e.g. S3, RDS)
- **For example**: Large objects or files should be stored in S3. The pointers and metadata associated with those files can be stored in SimpleDB. This allows you to quickly search for and access your files, while minimizing overall storage costs.
- Free: Data transferred between SimpleDB and other AWS within the same Region is free of charge.

DynamoDB

- DynamoDB is a fully-managed NoSQL database service
- Automatically replicates your data synchronously across multiple Availability Zones within an AWS Region to ensure high-availability and data durability.
- Designed for extremely high throughput and low latencies for both reads and writes
- Ability to scale to extremely large datasets while maintaining predictable performance
- Docs: DynamoDB
- Free: 100MB storage, and up to 5 writes/second and 10 reads/second of ongoing throughput capacity.
- Probably not feasible for the requirements of this project within free tier limits. Use at your own risk.

Relational Database Service (RDS)

- Fully featured relational database with automatic backups, patch management, and replication
- Native access to an SQL database which provides compatibility with existing tools and SQL databases.
- Ability to easily scale compute resources or storage capacity for each DB instance
- RDS can be managed from the management console.
- Note: Each DB Instance will run until termination, which occurs when you issue an API call to delete the DB Instance, or in the event of an instance failure
- Docs: Relational Database Service

Relational Database Service (RDS)

- **Update 2012-10-02**: RDS is now part of the Free Usage Tier.
- Enough hours to run a DB Instance continuously each month
- Resources are still limited, use them wisely.
- **Note**: Using the scale compute feature will consume capacity beyond the free tier usage capacity. You will incur standard, pay-as-you-go rates for this additional capacity.
- Free: 750 hours of RDS Micro DB Instance usage
- Free: 20GB storage, 10m I/Os, 20GB backup storage (includes user-initiated snapshots)
- Docs: RDS Free Usage Tier

Data Management Summary

- That's a lot of information, let's summarize
- Most bang for your $0:
  - Simple Storage Service (storage)
  - SimpleDB (database)
  - Relational Database Service (database)
- Useful: Amazon Database Summary

Scalability

Elastic Load Balancer (ELB)

- The ELB service is solely for managing incoming network traffic to your (multiple) EC2 instances
- An ELB can be setup in the management console.
- EC2 instances are added to an ELB to be managed
- Network traffic that reaches an ELB is distributed amongst its managed set of EC2 instances.
- An ELB will monitor its designated EC2 instances and only route new traffic to less used instances; that's why it's called a **load balancer**
- Docs: Elastic Load Balancing

Elastic Load Balancer (ELB)

- The ELB fulfills a vital role that you could implement yourself using some dedicated EC2 instances.
- Don't reinvent the wheel; use the ELB to do load balancing for you.
- ELBs are an important part of achieving a scalable cloud service, but an ELB does not scale your available resources.
- Instead, an ELB allows you to efficiently use your available resources.

ELB Usage Case

- An ELB can take incoming EKG data and route it to one of several **listener** instances who can process the data or submit it to a database.
- If one instance is receiving too much traffic, the ELB will try to route traffic elsewhere first.
- Without the ELB, one would need a dedicated **delegator** instance to achieve similar results.
- Without any **load balancing** at all, you would need to manually/automatically distribute users (EKG data submitters) amongst **listener** instances to try to distribute the burden equally.

Auto Scaling (AS)

- AS can be configured to react to the status of your EC2 services.
- AS can kill off or spawn new instances depending on how much load they are experiencing (how much they are being used).
- AS is critical to implement a scalable cloud service.
- AS only manages scalability of requested resources (instances), so other AWS are needed to make a truly scalable cloud service.
- AS is commonly used with ELB to provide scalability in resources and traffic handling.

Auto Scaling (AS)

- If you misconfigure AS, then you may accidentally spawn too many instances and exhaust your free tier usage for the month (and get billed)
- **For example**: 20 micro instances consume your monthly quota in less than 2 days
- AS is not available via the AWS management console.
- AS is available via the AWS command line tools
- AS may be available via AWS SDKs, but we don't know.
- Docs: Auto Scaling

Deployment & Management Services

- Docs: CloudWatch
- Docs: Virtual Private Cloud
- Docs: CloudFormation
- Docs: Elastic Beanstalk

AWS API (Tools) and SDKs

- Amazon Management Console provides limited control
- AWS API provides full control of AWS Services
- Docs: AWS General Reference
- Docs: AWS Command Line Tools
- Docs: Sample Code & Libraries

Live Demo

- Registration: CC, Phone
- Launching an Instance
- Security Groups, Keypairs (SSH)
- Basic EC2 Instance Monitoring
- Useful: Account Activity (detailed usage/charges)
- Useful: Account Overview
- Docs: Getting Started with Amazon EC2

Afterwards

- The AWS free usage tier will expire 12 months from the date you sign up.
- When your free usage expires, or if your application usage exceeds the free tier, you simply pay standard service rates
- There's more than just turning off any instances!
- Don't forget to delete everything in S3, EBS, etc.

Remember...


IF YOU FORGET TO DELETE EVERYTHING IN AWS
YOU'RE GONNA HAVE A BAD TIME
memegenerator.net