Week 6 Lecture 18

Applied

Helpful Resources

- http://aws.amazon.com/console/
- http://aws.amazon.com/developertools/351

What's in this lecture?

- AWS Console
- S3
- Cloudfront

AWS Console

- Web Interface for all of Amazon's web services
- Typical scenario:
  - client uses AWS console to get overview
  - you do work using ec2-tools
- Can fulfill every high level requirement to get your app in the cloud

What we'll focus on

- EC2
- S3
- Cloudfront

Amazon EC2: Instances

- An instance is the server your application is running on
- Typically not a dedicated, individual, physical server, but virtual
- Can be selected on axes:
  - CPU
  - Memory
  - Type (GPU)

Amazon EC2: Elastic IPs

- An IP address maps to your application server's physical location
- If the server moves, you have to update DNS records -- this can take a while!
- An 'elastic IP' is a fixed end-point you can set your IP address to point toward
- Behind the scenes, EC2 keeps that address always pointing to your server

Amazon EC2: EBS Volumes

- Imagine an infinitely long piece of paper you can cut into sheets of any any size
- This what EBS is to hard disk volumes
- Is a virtual filesystem formatted as you wish
- Warning: Can be susceptible to poor IO performance

Amazon EC2: EBS Snapshots

- Snapshotting is the process of dumping a volume and storing it in S3
- Think of it as EBS backups made at specific points in time
- Snapshot contains all written data:
  - Snapshot of 1 TB volume with 1 GB data will be 1 GB in size

Amazon EC2: AMIs

- ‘Amazon Machine Images’
- Copies of a server’s operating and filesystem
- Configured and built based on standard operating systems and installed libraries:
  - CentOS w/ a PHP stack
  - Ubuntu w/ a typical Ruby stack

Amazon S3

- ‘Simple Storage Service’
- Think of it as a gigantic key-value store that
  - scalable
  - eventually consistent
  - can store up to 5 TB sized objects

Why?

- You application needs:
  - to store lots of data
  - present that data quickly to requestors
  - a dead simple way to ensure that data is preserved and persistent

Amazon S3: Buckets

- Each Amazon account can create ‘buckets’
- Buckets are uniquely identified by their string name
- Can be thought of as the hash table name for your data

Amazon S3: Keys

- Keys map to the file objects you are storing
- Are also unique to the bucket
- Allows for HTTP access: http://s3.amazonaws.com/bucket_name/key

Amazon S3: Features

- Can set a time-to-live on an object
- Objects are replicated across servers and availability zones
- Can host an entire (static) website

Amazon Cloudfront

- Amazon's CDN (Content Delivery Network)
- Can map a domain to your S3 bucket
- Reduces latency to objects from all access points

Amazon Cloudfront: Application Example

- You have an iPhone app where users are globally distributed
- Each user *should* get the same fast access to content
- Solution:
  - put all application content into an S3 bucket
  - map to an Amazon Cloudfront Domain
  - Ship app with Cloudfront domain, hiding bucket

Exercises

- install and use the ec2-tools to go through this guide:

http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/