Auditing in the cloud

Ownership of data

- Historically, with the company
  - Company responsible to secure data
  - Firewall, infrastructure hardening, database security
- Auditing
  - Performed on site by inspecting processes and controls
  - Seizing data for investigation only after gaining company's permission
  - Company owning data is always in control of data (may not be secure)
- Storing data in cloud
  - Company shares responsibility with cloud service provider (CSP)
  - More responsibility with CSP higher up the cloud stack
  - Security and compliance become the core competencies for CSP
    * Securing and encrypting data, hardening environment, backup and recovery processes, other infrastructure-related tasks
  - Company still responsible to secure overall application
    * Security and compliance as shared responsibility
    * Auditing the entire solution becomes more complex
    * Auditing across multiple actors: consumers and providers

Data and cloud security

- Out of IT control? Out of security?
- Results from a recent study (Alert Logic, 2013)
  - Cloud is *not* inherently less safe than enterprise data centers
  - Attacks in CSP environments tend to be crimes of opportunity
    * Attacks in data centers tend to be more targeted and sophisticated
  - Web applications are equally threatened in cloud and enterprise data centers
- Study concluded that success rate for penetration from outside threats higher in corporate data centers

Auditing cloud applications

- Auditors validate that their clients adequately address a collection of controls and processes
- Different regulations to satisfy industry standards, business processes, and data requirements

  Physical environment Perimeter security and data center controls

  Systems and applications Security and control of network, databases, software

  Software development life cycle (SDLC) Deployment and change management

  Personnel Background checks, drug testing, security clearance

- Physical machine vs cloud
  - Controls and processes map to a CSP instead of an individual
  - Compliance a high priority in the cloud
    * Relying on information provided by CSP
  - Private cloud to retain total control of data and processes
- IaaS environment
  - Multitenant environment
  - Auditor of a tenant not allowed to access infrastructure to protect the rights of other tenants
  - IaaS provider's auditors audit perimeter security, processes, and controls
  - Client auditors forced to inspect audit reports from CSP to ensure compliance
  - For private cloud, auditors may have access to actual infrastructure
- PaaS environment
  - Physical aspects of auditing get more complex
  - Infrastructure as well as application stack abstracted and managed by CSP
    * Monthly patching, locking down OS, intrusion detection
    * Even DB may be managed and controlled by CSP; customer controls only DB access and administration of users
- SaaS environment
  - Even more responsibility outsourced to CSP
  - CSP responsible for entire application
- Importance of audit
  - Adherence to regulations for business processes in the cloud
  - HIPAA compliance for health care applications in the US
  - Out of compliance leads to fines, legal issues, lost business and bad publicity
- Important to understand responsibility for data in each service model

Regulations in the cloud

- Industry specific, type of data and transactions, standards for any cloud-based system
- Actors include
  - CSP
  - Company building the applications
- Infrastructure may be compliant but applications may not be
  - Entire application needs to pass the audit
- Regulations and controls table

Audit	Category	Description
ISO27001	Software	Computer system
SSAE-16	Security	Controls for finance, security, and privacy
Directive 95/46/ec	Security	European security and privacy controls
Directive 2002/58/ec	Security	European e-privacy controls
SOX	Financial	Public company financial accountability controls
PCI DSS	Credit card	Security and privacy of credit card information
HIPAA	Health	Security and privacy of health care information
FedRAMP	Security	Standards for cloud computing
FIPS	Software	Standard for computer systems
FERPA	Education	Security and privacy of education information

- Controls and processes for software best practices, security, and privacy
  - Incident management
  - Change management
  - Release management
  - Configuration management
  - Service level agreements
  - Availability management
  - Capacity planning
  - Business continuity
  - Disaster recovery
  - Access management
  - Governance
  - Data management
  - Security management
- Local laws
  - Country/state may have specific laws
  - Social media sites may not invest in passing various audits
    - Post terms and conditions that are accepted by users to use the services
    - If individual data is lost, there is not much he/she can do
  - Stricter adherence to regulations in B2B services
    - Loss of data not intended for public knowledge can be dangerous
    - Company's secrets, information on customers and partners, public relations problems
- Important for a CSP to provide audit certifications

Audit design strategies

- Identify all regulations that apply based on application requirements
  - Common regulation includes IT best practices regulation such as ISO 27001 standard and some security regulation such as SSAE-16 or SOC 2
- Additional regulations
  - Industry requirements (health care, government, education)

  - Data types (payments, personal identifiable information)
  - Location (country, transmission across country boundaries)

- Workstream in the product roadmap dedicated to auditing
  - Data management
  - Security management
  - Centralized logging
  - SLA management
  - Monitoring
  - Disaster recovery
  - SDLC and automation
  - Operations and support
  - Organizational change management

- Product evolution over time
  - Enterprise view of strategies to leverage the initial investment over future cloud applications in a consistent manner
  - Reduce maintenance costs and improve auditability
  - Add auditing requirements early in the application development stage
    - Part of the core application
    - Reduce risk and auditing costs

- Effect of chosen cloud model on amount of development required
  - For IaaS, cloud consumer has to share a large amount of responsibility
  - In private cloud, consumer has total responsibility for all necessary processes and controls
  - With public cloud, responsibility for infrastructure layer goes to CSP
  - With PaaS and SaaS, more responsibility shifts to CSP

- Audit vs speed to market, especially for startups

Data considerations in the cloud

Data characteristics

- Characteristics of data to consider
  - Physical characteristics
  - Performance requirements
  - Volatility
  - Volume
  - Regulatory requirements
  - Transaction boundaries
  - Retention period

- Two key decisions
  1. Multitenant or single tenant
  2. Type of data store: SQL, NoSQL, file, ...

Physical characteristics

- Location of data
  - Legal responsibilities
- Preexisting data or new data
  - Move preexisting data into cloud?
  - Create new data in cloud?
- Amount of data to be moved into cloud
  - Move data using offline storage
  - Risk of data being compromised during transportation
- Physical location of data
  - Legal aspects of physical location
  - Laws about transporting data across country/state boundaries
- Data ownership
  - Company building the software?
    - Search results from Google
  - Third party?
    - Navigation data for Google maps
  - Customer of the system?
    - Dropped pins on Google maps; documents in Google docs
- Data sharing with other parties
  - Hide any parts?
- Aspects involving privacy, security, and SLAs

Performance requirements

- Real-time performance
  - Subsecond response time
- Near real-time performance
  - Perceived real-time
    - Not really real-time but end-user cannot tell the difference
- Delayed time
  - A few seconds to batch time frame of daily, weekly, monthly, ...
- Faster response time will leverage memory over disk
- Common design patterns for high-volume fast-performing data sets
  - Use a caching layer
  - Reduce size of data sets

  - Separate databases into read-only and write-only nodes
  - Data segmentation into customer-, time-, or domain-specific segments
  - Archive aging data to reduce table sizes
  - Denormalize data sets

Volatility

- Frequency of change in data
- Static data sets
  - Event-driven data in chronological order
    - Web logs (page views, referring traffic, search terms, user IP addresses)
    - Transactions (bank debit/credit, point-of-sale purchases, stock trading)
    - Collections (readings from manufacturing machines, environmental readings like weather, human genome data)
  - Write-once/read-many type data sets
    - Mostly used for analysis over a period of time for patterns and behavior observation
  - Stored over long time periods; consume terabytes of space
  - Nonstandard DB practices to maximize performance
    - Denormalize data, leverage star or snowflake schemas, NoSQL databases, big data technologies
- Dynamic data sets
  - Frequently changing data
  - Normalized relational DBMS
    - Good for processing ACID transactions (atomicity, consistency, isolation, durability)
    - Ensure data reliability
    - Protect integrity of data by ensuring that duplicate data and orphan records do not exist
  - Speed of data flow (add/change/delete)
  - Understanding different disk storage systems
    - On AWS, S3 is highly reliable but not best performing
    - EBS volumes are high performing local disk systems but lack the reliability and redundancy of S3

Volume

- Amount of data to maintain and process
- Performance of relational DBMS
  - Slow and expensive to maintain beyond a certain amount of data
- Amount of data to be maintained and accessible online vs archived
- Backup strategy
  - Frequency of full and incremental backup
  - Perform backups on a slave database so as to not impact application performance

Regulatory requirements

- Certifications in various regulations
- Data encryption in flight and at rest, especially for data classified as PII (personally identifiable information)
  - Performance overhead
  - Even bigger issue for highly volatile data
  - May contribute to choice for private cloud

Transaction boundaries

- Unit of work on the web
- Process flow from beginning to end of transaction
  - Booking flight, hotel, car rental on Expedia
- Data points to store state
  - RESTful services (Representational State Transfer) are stateless by design
  - Architect needs to determine a way to save state for multipart transaction
    * Caching, writing to queue, or writing to temporary table or disk
- Frequency of multipart transactions (disk vs cache)

Retention period

- How long to keep data
  - Financial data stored for seven years for audit purposes
  - Bank statements available online from six months to a year
    * Older statements can be requested
    * Requests handled in batch and may incur a fee

Multitenant or single tenant

- Determined by data characteristics
- Multitenancy in data layer of architecture
  - Multiple organizations or customers share a group of servers
  - Master-slave configuration of servers to support a tenant
- Single tenant
  - Only one tenant on a group of servers
- Total isolation
  - Applications and data isolated on their respective servers
  - Both database layer and application layer have dedicated resources for each tenant
  - Advantages: Independence, privacy, highest scalability

  - Disadvantages: Most expensive, minimal reuse, highest complexity
  - Applications must be infrastructure aware and know how to point to correct infrastructure
  - Useful when tenant has enormous amount of traffic
  - Dedicated servers maximize scaling while avoiding disruptions for other clients
- Data isolation
  - Application takes a multitenant approach to the application layer by sharing application servers, web servers, and other services
  - Database layer is single tenant
  - Advantages of independence and privacy while reducing some costs and complexities
  - Protects the proivacy of each tenant's data and allows tenants to scale independently
  - Amount of traffic is not overwhelming but there is a need to store data in its own schema for privacy reasons
- Data segregation
  - Separate tenants into different database schemas sharing the same servers
  - All layers are shared for all tenants
  - Advantages: Most cost effective, least complex, highest reuse
  - Disadvantages: Lack of independence, lowest performance, lowest scalability
  - Performance issues with one tenant can create issues for other tenants

Choosing data store types

- Relational databases
  - Have been around for long
  - Good for online transaction processing (OLTP) applications
  - Guarantee that transactions are processed successfully to store data in database
  - Superior security features
  - Powerful query engine
  - Enforce referential integrity
    * Accomplished by a lot of overhead built into database engine
    * Ensure that transactions complete and committed before data is stored into database
  - Require indexes to assist in retrieval of records
    * With increasing size, indexes become counterproductive
- NoSQL databases
  - Can handle increasing amount of data
  - Provide access to elastic cloud resources
  - Falling costs of disk resources
  - Useful for analytics, data mining, pattern recognition, and machine learning
- Four types of NoSQL databases
  1. Key-value store
     - Simplest NoSQL database type
     - Hash table

  - Unique key with a pointer points to a particular data item
  - Fast and highly scalable
    * Good for processing massive amounts of writes such as tweets
  - Good for reading large, static-structured data such as historical orders, events, and transactions
  - No schema
    * Bad choice to handle complex data and relationships
  - Redis, Voldemort (LinkedIn), DynamoDB (Amazon)

2. Column store
   - Store and process large amount of data distributed over many machines
   - Hash key points to multiple columns organized in column families
   - Columns can be added on the fly and do not have to exist in every row
   - Incredibly fast, scalable, and easy to alter on the fly
   - Good to integrate data feeds from different sources with different structures
   - Not good for interconnected data sources
   - Hadoop, Cassandra
3. Document store
   - Used to store unstructured data
     * XML, JSON, PDF, Word, Excel
   - Logging solutions to combine log files from different sources
     * Database logs, web server logs, application server logs, application logs
   - Good at scaling large amount of data in different formats
   - Not good with interconnected data
   - CouchDB, MongoDB
4. Graph database
   - Used to store and manage interconnected relationships
   - Visual representation of relationships, especially in social media analysis
   - Good at graphing
   - Not good at other things because entire relationship tree must be traversed to produce results
   - Neo4j, InfoGrid

- Other storage options
  - Data stored as files
    * Photos, videos, MP3
  - Content delivery network
    * Network of distributed computers located in multiple data centers
    * High availability and high performance
    * Good for streaming media and other bandwidth intensive data