Disaster Recovery Planning

- Failure in technology
  - Web server, database server, data center
  - Expect every component to fail and design for the failure
  - Major catastrophic events classified into human, on purpose or by accident, (9/11); natural (Hurricane Katrina or earthquake); or technological failures
  - Plan for physical security and human contingency using evolving technologies
    - Company operations must go on during, and after, a disaster
    - Workforce resilience
      - Allow employees to work remotely during a disaster using VPN
      - Open communications and emergency notification systems
      - Support from emergency management; setting up internet cafes and charging stations in the event of power failure
      - Use of social media (Facebook/Twitter) to manage corporate communications and controlling rumors
      - May even provide cots, flashlights, food, and water for employees who stay in office and have a remote recovery site in operation to restore critical systems quickly
    - Some of the rules are mandated by federal laws; others are needed just to stay in business
  - Risks mitigated due to virtualization and the ability to run multiple live data centers with active failover
    - Reduction in time between system failures and data recovery points

Cost of downtime

- Getting a solution to marketplace vs deploying a failsafe solution
- Strategies for disaster recovery
  1. Understand three important variables from a business perspective
     (a) Recovery time objective (RTO)
       - Time within which business requires that the service is back up and running
       - Possibly five minutes or less for an e-commerce site
       - Reporting system can tolerate longer down time because of no impact on revenue or customer satisfaction

     (b) Recovery point objective (RPO)
       - Amount of time in which data loss can be tolerated
       - Parts dealing with financial transactions must have zero or near-zero tolerance for data loss
       - Social aspects of an e-commerce site can tolerate longer down time

     (c) Value placed on recovery
       - Measurement of worth to the company to mitigate disaster situations
       - Digital incentive platform for a small business (downtime of an hour or two is acceptable) vs big retailers (requires fully redundant virtual data centers across multiple availability zones)
       - Criticality of service (health and safety of citizens)
       - Service reliability (streaming music)

Disaster recovery strategies for IaaS

- Complex as the CSC is responsible for the application stack

- For public IaaS, CSC depends on CSP to manage physical data center

- Preventing disasters in Amazon cloud
  - Amazon cloud consists of regions and availability zones
    - Regions located across the globe
    - Zones are independent data centers within a region
  - Typical outage occurs within a single availability zone
    - Build redundancy across multiple zones to maintain uptime even when AWS has outage
  - An API may have outage impacting multiple zones
    - Amazon Elastic Block Store (EBS)is a service to provide network attached disks to install databases
    - If EBS has issues across zones, cross-zone redundancy would not prevent system from failing
  - Redundancy across regions
    - More complex and expensive than cross-zone redundancy
    - Moving data across zones
      - Incurs extra charges
      - Introduces extra latency
    - Cost and complexity of cross-region redundancy must be balanced with the value of recovery, RTO and RPO

- Hybrid cloud solution
  - Leverage a private cloud provider that supports Amazon's API
  - Restrict AWS API usage to just the APIs that are supported by private cloud vendor if all parts of the system need to be recovered
  - Private cloud in the hybrid cloud creates another availability zone with the APIs in the private zone isolated from any issues in AWS

- Leverage multiple cloud vendors
  - Build system to not lock into an IaaS vendor
  - Do not use proprietary APIs to be *cloud agnostic*
  - Isolate vendor-specific APIs and build logic to execute appropriate API based on vendor

Recovery in primary data center

- Standard set of best practices to recover the database from a disaster
  1. Classic backup and restore method
     - Create daily full backups and incremental backups
     - Store backups into a disk service provided by cloud vendor
     - Copy backups to a secondary data center and to some third-party vendor
     - Database goes offline, gets corrupted, or any other issue
       - Restore last good full backup and apply incremental backups
     - Cheapest solution with no redundant servers
     - RTO is long as database cannot be brought back online until backups restored and data quality verified
  2. Redundant data centers – active-passive cold
     - Secondary data center prepared to take over duties from primary data center
     - *Cold* – Redundant servers are not on and running

    - Set of scripts ready to run in case of emergency to provision a set of servers configured exactly the same as primary data center
    - Restore from the latest backup in the event of emergency
    - Cost-effective way to deal with outage as *cold* servers do not cost anything unless provisioned
    - Not acceptable if RTO is less than a few minutes
  3. Redundant data centers – active-passive warm
     - Runs the database server *hot*
       - Always on and always in sync with the master data center
     - Other servers are *cold* and provisioned upon execution of disaster recovery plan
     - More expensive than active-passive cold
     - Greatly reduces downtime as no database restore required
     - Hot database can be allocated for other uses instead of waiting for disaster declaration
       - Use for business intelligence workloads
     - Useful for systems with a low RPO
  4. Redundant data centers – active-active hot
     - Fully redundant data centers at all times
     - Complete failure of one data center causes no downtime at all
     - Provides low tolerance for lost data and downtime
     - High value of recovery; very low impact to customers
     - Database uses master-slave replication across data centers
     - If primary data center fails, the secondary data center becomes the new master
     - When failed data center recovers, downed databases start to sync up
     - When all data is synced, control goes back to primary data center to act as master again
     - Failure is not an option

Disaster recovery strategies for PaaS

- Public PaaS
  - Entire platform, including application stack and infrastructure, is responsibility of vendor
  - Abstract away all the work to handle underlying infrastructure and application stack, including scaling databases, designing for fail over, and patching servers
  - Developers focus on business requirements
  - Consumer responsible for applications built on top of platform
  - In emeregency, consumer at the mercy of vendor's disaster recovery plan
- Private PaaS
  - Vendor abstracts the development platform
  - Installing and managing application stack becomes simple but consumer has to manage the infrastructure
  - Consumer back in control in case of emergency

Disaster recovery strategies for SaaS

- Disaster recovery plan for use case where an SaaS service is unavailable for an extended period
  - SaaS-based financial system offline for a week
  - Typically, customer dependent on the SaaS provider without much recourse
- Minimally, SaaS contract from the vendor should have a software escrow
  - Protects the buyer if SaaS vendor goes out of business, or voids the contract if purchased by another company
  - Escrow holds the vendor's IP in an independent third party's holding area, giving the buyer ownership of data