Designing Cloud-based Solutions

Outline

- Design considerations for a cloud-based solution

Accessibility

- Maximise user access
  - Public solution (e.g.: consumer website)
    - Making great marketing sense
    - Required by law
- Lock down the system and control
  - Secure site
  - Access specific features
  - Ensuring the security of the login process

Audit

- Internal control:
  - Solution's processing is correct
  - Free from outside manipulation
  - Active control:
    - Generate a processing exception
  - Passive control:
    - Logging events
    - Snapshots

Availability

- SLA: guarantee system availability
  - % of uptime
  - Ex:
    - 99.9% uptime, is it acceptable?
- Identify system's requirements
  - Using redundant colocated servers

Backup

- Data and databases:
  - Solutions
  - Method Impact
- Redundant data-storage solution:
  - Cost vs. Risk
- Third party solution:
  - Backup policies and procedures
  - Integrating

Existing and Future Capacity

- Capacity planning:
  - Monitor applications
  - Know system's resource use
    - User demand, CPU utilisation, RAM use, data-storage consumption, etc.
- Design for scalability
  - Vertical scaling
  - Horizontal scaling

Configuration Management

- Cloud-based solution:
  - Any time, any place, any device
  - Variety of OS, browsers, device-specific GUIs
    - Ex: security issues, new versions, etc.
- Configuration solutions
  - Top layer
  - Third party (SaaS):
    - Company's patch management, policies and procedures

Deployment

- Desktop virtualisation:
  - OS on demand
  - Myriad of ways to deploy a system
- Deployment Solution
  - Potential user type and its environment attributes
  - How to deploy initial solution, system upgrades

Disaster Recovery

- Reducing the risk from disaster
- Disaster recovery and business continuity
  - Risks vs. costs
  - Business impact
  - Affordable solution
  - Cloud's resources: flexibility

Environment

- Green-computing
  - Environmentally friendly
- Data centre
  - Power consumption
  - Air conditioner
  - Migrate to PaaS, IaaS providers: efficiency?

Interoperability

- Wide range of application
  - CRM, HR application, etc.
- Integrate the solutions
- Share data across solution
  - Buy and install middleware
  - Cloud-based middleware

Maintainability

- System maintenance phase
  - Most costly
- Code reuse, code maintainability
  - Highly functional
  - Independent
- SaaS
  - Long-term nature

Performance

- Optimising what you have
  - Reduce the use of graphics on key pages
  - Optimise the graphics file format for all images
  - Compress large text blocks before downloading
  - Utilise data and application caching
  - Fine-tune disk and database I/O operations
  - Reduce network operations
  - Fine-tune secure data communication transactions

Privacy

- Protect users' data privacy
  - Healthcare
  - E-commerce
  - …
- External access
- Internal access
- Backing up / Replicating

Portability

- The ease with which a solution can be moved
  - Different cloud providers
- Open source tools
  - Application's portability
- Provider specific API
  - Vendor lock-in

Recovery, Reliability & Robustness

- Recover from common events
  - Server failure
  - User error
  - Power outage
- Identify potential signal points of failure
- Backup design, system redundancy design
- Robustness: ability to continue operations
  - Monitor system resources
  - Alert administrators

Response Time

- Fast system response
  - Page download times
  - System response time: users' operation
- Site's capacity plan design
- Testing user experience
  - Different geographic locations
  - Different connection speeds
  - Variety of browsers

Security

- Software:
  - Installation and version management
- Early awareness of security incidents
- Data privacy
- Jurisdictional issues
- Multi-tenant solution issues
- Cloud-provider failure or collapse
- Defense mechanisms for network attacks
- Data wiping for share-storage space
- Physical security considerations

Testability and Usability

- How to test various aspects of the design
  - Functional requirements
  - Non-functional requirements
- Ease of use
- Prototype

Creating a Cloud-based application

- Building a web-based application
- Programming language
  - PHP, Ryby, Perl, Python, C#, etc.
  - HTML, CSS
  - Databases: MySQL…
- PaaS
  - Tools to build and deploy solutions
- Yahoo!Pipes, Google App Engines, Windows Azure, etc.