Cloud Computing

Cloud Computing
Technologies and Types

Dell Zhang
Birkbeck, University of London
2018/19
The Technological Underpinnings of Cloud Computing

• Data centres (see former lectures)
• Virtualisation (see former lectures)
• RESTful APIs (see later lectures)
• Cloud storage (see former lectures)
• NoSQL databases (see later lectures)
• Elasticity (see former lectures)
Understanding the Different Classifications of Clouds

Cloud Enablement
Infrastructure and utilities that provide the glue necessary to run the system

Software as a Service (SaaS)
Packaged software application

Platform as a Service (PaaS)
Environment for building a managed application with an IDE with a rich class library that executes in a runtime container

Infrastructure as a Service (IaaS)
Environment for building a native application
X as a Service

• Infrastructure as a Service (IaaS)
  – Why buy machines when you can rent cycles?
  – Examples: Amazon EC2, Rackspace

• Platform as a Service (PaaS)
  – Give me nice API and take care of the maintenance, upgrades, ...
  – Example: Google App Engine (GAE)

• Software as a Service (SaaS)
  – Just run it for me!
  – Example: Gmail, Salesforce’s Online CRM
An IaaS supplies virtual machine images of different operating system flavours.

- These images can be tailored by the developer to run any custom or packaged application.
- The user can bring online and use instances of these virtual machine images when needed.
PaaS

• A PaaS abstracts away the interaction with the bare metal of the system, and thus lets software developers concentrate specifically on writing the application.
  – You don’t need to directly administer the virtual operating systems.
  – This simplification generally comes at the cost of less flexibility and the requirement to code in the specific languages supported.
SaaS

• An SaaS provides software applications that are available on an on-demand basis.
Serverless Computing

• In the majority of [cloud computing] cases,
  – the user must *deploy* VMs (either directly or indirectly) to support these capabilities and *delete* the VMs when they are no longer needed.

• At times, however, this overhead is unacceptable.
  – When you want an action to take place in response to a relatively rare event, the cost of keeping a VM running continuously so that a program can wait for the event may be unacceptably high.
Serverless Computing

• For example
  – the user may wish to perform some bookkeeping when a new file is created in a cloud repository
  – The user may wish to receive a notification when an important event occurs
Serverless Computing

• In serverless computing,
  – the user provides a simple function to be executed under certain conditions;
  – the cloud provider keeps a set of machines running to execute these functions on the user’s behalf;
  – the user is charged only for the execution of the task, not for maintaining the servers.
FaaS

• Most serverless vendors offer compute runtimes aka Function as a Service (FaaS) platforms, which execute application logic but do not store data.
  – AWS Lambda, introduced by Amazon in 2014, was the first public cloud infrastructure vendor with an abstract serverless computing offering.
  – Google Cloud Functions
  – Microsoft Azure Functions
Cloud Types

Hybrid

Private/ Internal

Public/ External

The Cloud

On Premises / Internal

Off Premises / Third Party

Cloud Computing Types
Cloud Types

• Public Cloud
  – The cloud made available to the general public by a service provider, either free or offered on a pay-per-use model.

• Private Cloud
  – The cloud operated solely for a single organization and not shared with others.

• Community Cloud
  – The cloud shared between several organizations from a specific community with common concerns.
    – such as ...?
Cloud Types

• Hybrid Cloud
  – A composition of two or more different types of clouds.
  – Why?
    • (1) The organisation may want to host some critical, secure applications in their private cloud, but use a public cloud for the not-so-critical applications.
    • (2) Cloud bursting: the organisation uses its own cloud for normal usage, but an external cloud is used for peak loads when excess capacity is needed.
To Be Private, or Not?

• Private clouds are at small scale
  – Big cost savings are driven by huge volume
• Legacy applications don’t cloudify easily
  – You can achieve only marginal improvements without re-architecting them
• On-premises doesn’t mean more secure
  – Unless your company spends more money and energy thinking about security than Amazon etc.
• Do what you do best
  – Private clouds will always be many steps behind public clouds in the rate of innovation and optimisation
Matching Cloud Providers to Your Needs

• Amazon Web Services (AWS)
• Google App Engine (GAE)
• Windows Azure
AWS

• IaaS + PaaS
Global infrastructure

Regions
An independent collection of AWS resources in a defined geography
A solid foundation for meeting location-dependent privacy and compliance requirements
Global infrastructure

Availability Zones

- Designed as independent failure zones
- Physically separated within a typical metropolitan region

AWS Global Infrastructure
Global infrastructure

Edge Locations

To deliver content to end users with lower latency
A global network of edge locations
Supports global DNS infrastructure (Route53) and Cloud Front CDN
Networking

Direct Connect

Dedicated connection between your IT infrastructure and the AWS datacenters
Extend your network infrastructure and VLANs into AWS
VPN Connection

A Hardware VPN connection connects Amazon environment to your datacenter

Internet Protocol security (IPsec) VPN connection

Commonly used hardware supported
Virtual Private Cloud

Private, isolated section of the AWS Cloud
Launch resources in a virtual network that you define
complete control over your virtual networking environment
Route 53

**Highly available and scalable Domain Name System**

**Extremely reliable and cost effective**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Supported from AWS global edge locations for fast and reliable domain name resolution</td>
</tr>
<tr>
<td>Scalable</td>
<td>Automatically scales based upon query volumes</td>
</tr>
<tr>
<td>Latency based routing</td>
<td>Supports resolution of endpoints based upon latency, enabling multi-region application delivery</td>
</tr>
<tr>
<td>Integrated</td>
<td>Integrates with other AWS services allowing Route 53 to front load balancers, S3 and EC2</td>
</tr>
<tr>
<td>Secure</td>
<td>Integrates with IAM giving fine grained control over DNS record access</td>
</tr>
</tbody>
</table>
Elastic Compute Cloud (EC2)

Basic unit of compute capacity
Range of CPU, memory & local disk options
13 Instance types available, from micro to cluster compute

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexible</td>
<td>Run windows or linux distributions</td>
</tr>
<tr>
<td>Scalable</td>
<td>Wide range of instance types from micro to cluster compute</td>
</tr>
<tr>
<td>Machine Images</td>
<td>Configurations can be saved as machine images (AMIs) from which new instances can be created</td>
</tr>
<tr>
<td>Full control</td>
<td>Full root or administrator rights</td>
</tr>
<tr>
<td>Secure</td>
<td>Full firewall control via Security Groups</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Publishes metrics to Cloud Watch</td>
</tr>
<tr>
<td>Inexpensive</td>
<td>On-demand, Reserved and Spot instance types</td>
</tr>
<tr>
<td>VM Import/Export</td>
<td>Import and export VM images to transfer configurations in and out of EC2</td>
</tr>
</tbody>
</table>
Compute

Auto-scaling

Automatic re-sizing of compute clusters based upon demand

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>Define minimum and maximum instance pool sizes and when scaling and cool down occurs</td>
</tr>
<tr>
<td>Integrated to CloudWatch</td>
<td>Use metrics gathered by CloudWatch to drive scaling</td>
</tr>
<tr>
<td>Instance types</td>
<td>Run auto scaling for on-demand instances and spot. Compatible with VPC</td>
</tr>
</tbody>
</table>
Elastic Load Balancing

Create highly scalable applications
Distribute load across EC2 instances in multiple availability zones

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-scaling</td>
<td>Automatically scales to handle request column</td>
</tr>
<tr>
<td>Available</td>
<td>Load balance across instances in multiple availability zones</td>
</tr>
<tr>
<td>Health checks</td>
<td>Automatically checks health of instances and takes them in or out of service</td>
</tr>
<tr>
<td>Session stickiness</td>
<td>Route requests to the same instance</td>
</tr>
<tr>
<td>Secure sockets layer</td>
<td>Supports SSL offload from web and application servers with flexible cipher support</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Publishes metrics to Cloud Watch</td>
</tr>
</tbody>
</table>
# Storage

## S3 - Durable storage, any object

99.999999999% durability of objects

Unlimited storage of objects of any type

Up to 5TB size per object

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexible object store</td>
<td>Buckets act like drives, folder structures within</td>
</tr>
<tr>
<td>Access control</td>
<td>Granular control over object permissions</td>
</tr>
<tr>
<td>Server-side encryption</td>
<td>256bit AES encryption of objects</td>
</tr>
<tr>
<td>Multi-part uploads</td>
<td>Improved throughput &amp; control</td>
</tr>
<tr>
<td>Object versioning</td>
<td>Archive old objects and version new ones</td>
</tr>
<tr>
<td>Object expiry</td>
<td>Automatically remove old objects</td>
</tr>
<tr>
<td>Access logging</td>
<td>Full audit log of bucket/object actions</td>
</tr>
<tr>
<td>Web content hosting</td>
<td>Serve content as web site with built in page handling</td>
</tr>
<tr>
<td>Notifications</td>
<td>Receive notifications on key events</td>
</tr>
<tr>
<td>Import/Export</td>
<td>Physical device import/export service</td>
</tr>
</tbody>
</table>
Elastic Block Store

High performance block storage device
1GB to 1TB in size
Mount as drives to instances

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>High performance file system</td>
<td>Mount EBS as drives and format as required</td>
</tr>
<tr>
<td>Flexible size</td>
<td>Volumes from 1GB to 1TB in size</td>
</tr>
<tr>
<td>Secure</td>
<td>Private to your instances</td>
</tr>
<tr>
<td>Available</td>
<td>Replicated within an Availability Zone</td>
</tr>
<tr>
<td>Backups</td>
<td>Volumes can be snapshotted for point in time restore</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Detailed metrics captured via Cloud Watch</td>
</tr>
</tbody>
</table>
Relational Database Service

Database-as-a-Service

No need to install or manage database instances

Scalable and fault tolerant configurations

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform support</td>
<td>Create MySQL, SQL Server and Oracle RDBMS</td>
</tr>
<tr>
<td>Preconfigured</td>
<td>Get started instantly with sensible default settings</td>
</tr>
<tr>
<td>Automated patching</td>
<td>Keep your database platform up to date automatically</td>
</tr>
<tr>
<td>Backups</td>
<td>Automatic backups and point in time recovery and full DB backups</td>
</tr>
<tr>
<td>Backups</td>
<td>Volumes can be snapshotted for point in time restore</td>
</tr>
<tr>
<td>Failover</td>
<td>Automated failover to slave hosts in event of a failure</td>
</tr>
<tr>
<td>Replication</td>
<td>Easily create read-replicas of your data and seamlessly replicate data across availability zones</td>
</tr>
</tbody>
</table>
**DynamoDB**

*Provisioned throughput NoSQL database*
*Fast, predictable performance*
*Fully distributed, fault tolerant architecture*

---

**Feature** | **Details**
--- | ---
**Provisioned throughput** | Dial up or down provisioned read/write capacity
**Predictable performance** | Average single digit millisecond latencies from SSD backed infrastructure
**Strong consistency** | Be sure you are reading the most up to date values
**Fault tolerant** | Data replicated across availability zones
**Monitoring** | Integrated to Cloud Watch
**Secure** | Integrates with AWS Identity and Access Management (IAM)
**Elastic MapReduce** | Integrates with Elastic MapReduce for complex analytics on large datasets
Amazon SQS

Reliable, highly scalable, queue service for storing messages as they travel between instances

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliable</td>
<td>Messages stored redundantly across multiple availability zones</td>
</tr>
<tr>
<td>Simple</td>
<td>Simple APIs to send and receive messages</td>
</tr>
<tr>
<td>Scalable</td>
<td>Unlimited number of messages</td>
</tr>
<tr>
<td>Secure</td>
<td>Authentication of queues to ensure controlled access</td>
</tr>
</tbody>
</table>
**Application Services**

**Simple Workflow**

Reliably coordinate processing steps across applications
Integrate AWS and non-AWS resources
Manage distributed state in complex systems

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process state</td>
<td>Maintain application state across complex workflows in a reliable and available manner</td>
</tr>
<tr>
<td>Tracking</td>
<td>Tracks executions and log process for audit purposes</td>
</tr>
<tr>
<td>Consistency</td>
<td>Ensures processing tasks are executed and duplicity of events does not occur</td>
</tr>
<tr>
<td>Simple</td>
<td>Simple Decider and Task programming model for rapid integration</td>
</tr>
</tbody>
</table>
Cloud Search

Elastic search engine based upon Amazon A9 search engine
Fully managed service with sophisticated feature set
Scales automatically

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-scaling</td>
<td>Automatically scales based upon request volumes and data volumes</td>
</tr>
<tr>
<td>High performance</td>
<td>In memory operation means consistently low latency for search results</td>
</tr>
<tr>
<td>Sophisticated features</td>
<td>Support for faceting, stemming, synonyms, stop words and custom rank expressions</td>
</tr>
<tr>
<td>Low cost</td>
<td>Elastic service, pay for what you use</td>
</tr>
</tbody>
</table>
CloudFront

World-wide content distribution network
Easily distribute content to end users with low latency, high data transfer speeds, and no commitments.
Elastic Beanstalk

One-click deployment from Eclipse, Visual Studio and Git

Rapid deployment of applications

All AWS resources automatically created

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform support</td>
<td>Containers for Java, .net and PHP</td>
</tr>
<tr>
<td>Resource creation</td>
<td>Creates load balancer, instances, autoscaling and monitoring automatically</td>
</tr>
<tr>
<td>Monitoring &amp; Logs</td>
<td>Integrated with Cloud Watch and consolidates server logs</td>
</tr>
<tr>
<td>Versioning</td>
<td>Manage versions of applications and easily rollback deployments</td>
</tr>
<tr>
<td>Notifications</td>
<td>Receive alerts on key events</td>
</tr>
<tr>
<td>Full resource access</td>
<td>Access all underlying AWS resources as necessary</td>
</tr>
</tbody>
</table>
This template creates a CloudFormation stack that uses Amazon CloudFront and an Amazon EC2 AMI for Adobe Flash Media Server 4.1 to enable HTTP streaming of your live event.

Cloud Formation

Automate creation of ‘stacks’ in a repeatable way
Scripting framework for AWS resource creation

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform support</td>
<td>Support for AWS resources from EC2 to IAM</td>
</tr>
<tr>
<td>Resource creation</td>
<td>Creates AWS resources behind the scenes and reports on progress</td>
</tr>
<tr>
<td>Declarative</td>
<td>Specify stacks in JSON format and source control your environments</td>
</tr>
<tr>
<td>Customizable</td>
<td>Drive stack creation with parameters</td>
</tr>
</tbody>
</table>
Identity & Access Management
Granular control of user rights with AWS
Automated granting of EC2 service rights

Software Developer Kits
Comprehensive support of programming models for using AWS services
+ others
Simple Email Service
Simple Notification Service
ElastiCache
Elastic MapReduce
CloudWatch
...
Choose AWS

• If you
  – want to use third-party open-source software
  – have existing code
  – want to transfer a web app to your own machine/servers later
  – port code to another language
  – want complete control
  – need to stress/load test an app
    • for example, load up 1,000 instances
GAE

• PaaS
  – Languages and runtime environments
    • Python
    • Java
    • Go
  – Sandbox
  – Datastore
GAE vs AWS

GAE:
• Higher-level functionality (e.g., automatic scaling)
• More restrictive (e.g., respond to URL only)
• Proprietary lock-in

EC2/S3:
• Lower-level functionality
• More flexible
• Coarser billing model

Python/Java
BigTable
Other API’s

VMs
Flat File Storage
Choose GAE

• If you
  – have no pre-existing code
  – are building request-response web apps or mashups
  – consider time-to-market the most important thing
  – aren’t doing anything fancy (installing software)
  – aren’t worried about lock-in to Google
Windows Azure

- IaaS + PaaS
Choose Azure

• If you
  – already use the .NET and SQL Server portions of the Microsoft stack
  – have existing code developed to those Microsoft APIs
  – have teams that normally develop in Visual Studio using C#
  – want to blend development from desktop to cloud
  – have no issue with lock-in to Microsoft
Take Home Messages

• Understanding the Different Classifications of Clouds
  – IaaS, PaaS, SaaS
  – Public, Private, Community, Hybrid

• Matching Cloud Providers to Your Needs
  – Amazon, Google, Microsoft