CS415 Module 9 Part B - Virtual Servers

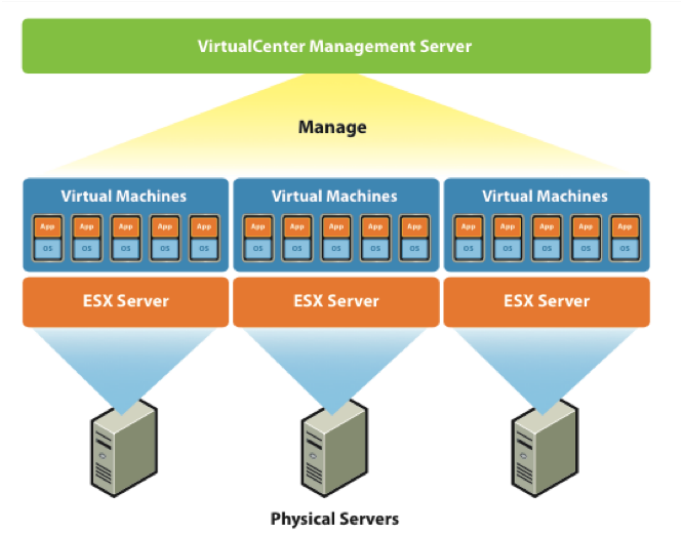
Athens State University

Outline

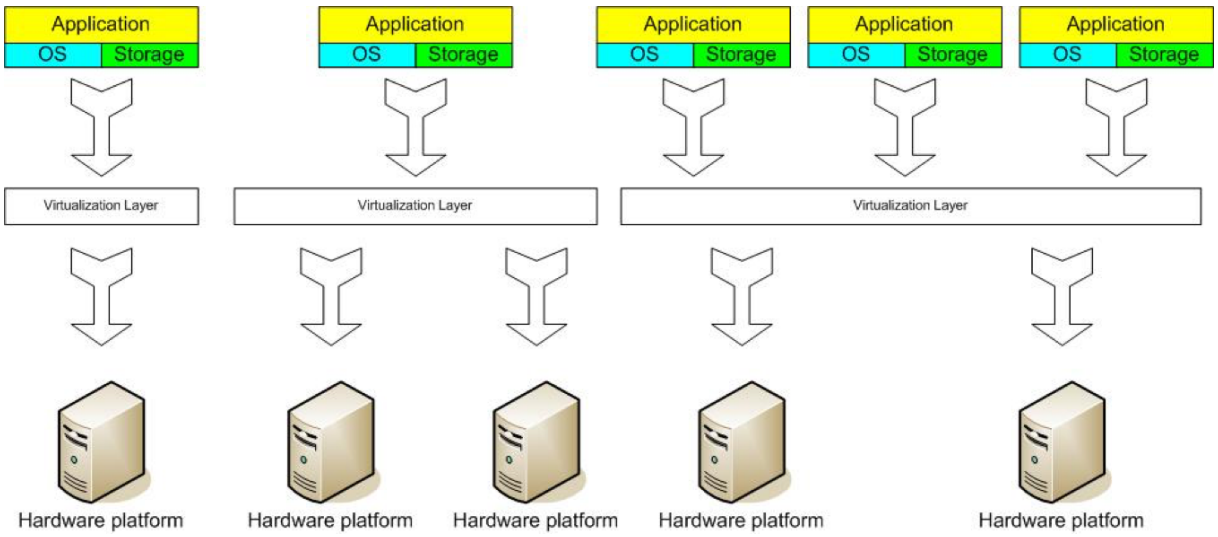
Contents

1 Virtual Infrastructure

Data Center Consolidation



The Virtual Server Concept



The Virtual Server concept

- Virtual servers seek to encapsulate the server software away from the hardware This includes the OS, the applications, and the storage for that server.
- Servers end up as mere files stored on a physical box, or in enterprise storage.
- A virtual server can be serviced by one or more hosts, and one host may house more than one virtual server.

The Virtual Server Concept

- Virtual servers can still be referred to by their function i.e. email server, database server, etc.
- If the environment is built correctly, virtual servers will not be affected by the loss of a host.
- Hosts may be removed and introduced almost at will to accommodate maintenance.

The Virtual Server Concept

- Virtual servers can be scaled out easily.
  - If the administrators find that the resources supporting a virtual server are being taxed too much, they can adjust the amount of resources allocated to that virtual server
- Server templates can be created in a virtual environment to be used to create multiple, identical virtual servers
- Virtual servers themselves can be migrated from host to host almost at will.

The Virtual Server Concept

- Pros
  - Resource pooling
  - Highly redundant

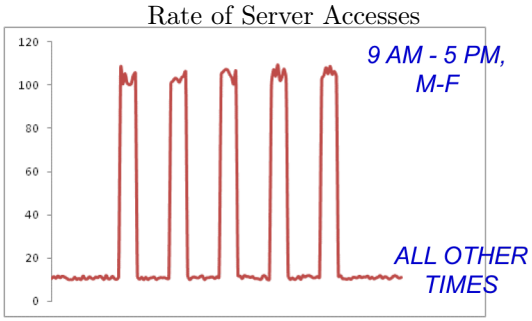
  - Highly available
  - Easy to quickly deploy new servers
  - Reconfigurable while services are running
  - Optimizes physical resources by doing more with less

- Cons
  - Slightly harder to conceptualize
  - Slightly more costly (hardware, OS, apps, and abstraction layer)

2 Cloud Computing

Suppose you are Forbes.com

- You offer on-line real time stock market data
- Why pay for capacity on weekends or overnight



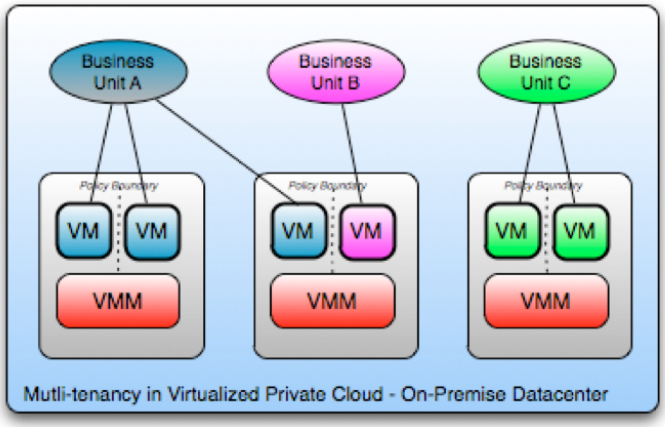
The Solution

- Host your web site in Amazon's EC2 *Elastic Compute Cloud*
- Provision new servers every day, and de-provision them every night
- Pay just ten cents per server per hour (more for higher capacity servers)
- AND let Amazon have all of the hardware worries!

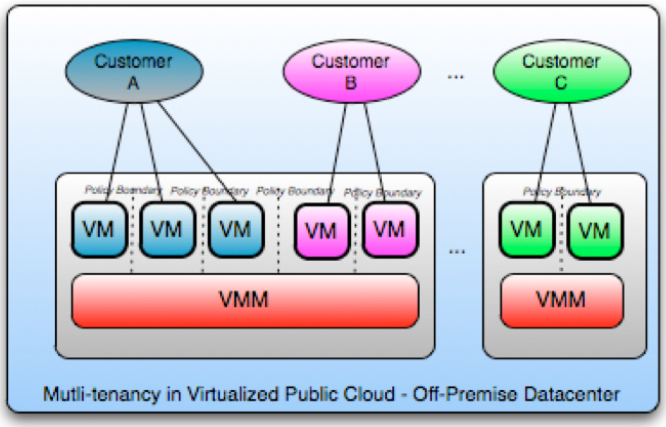
Cloud Computing: Enabled By Virtualization

- You don't have own the hardware, you "rent" it as needed from a cloud
- There are public clouds
  - Two most well-known examples are Amazon's EC2 and Microsoft's Azure cloud services
  - Now major focus for both companies
- Or a company can create a private cloud
  - With more control over security, etc.
  - Now very common within the DoD and its contractors

Private vs. Public Cloud



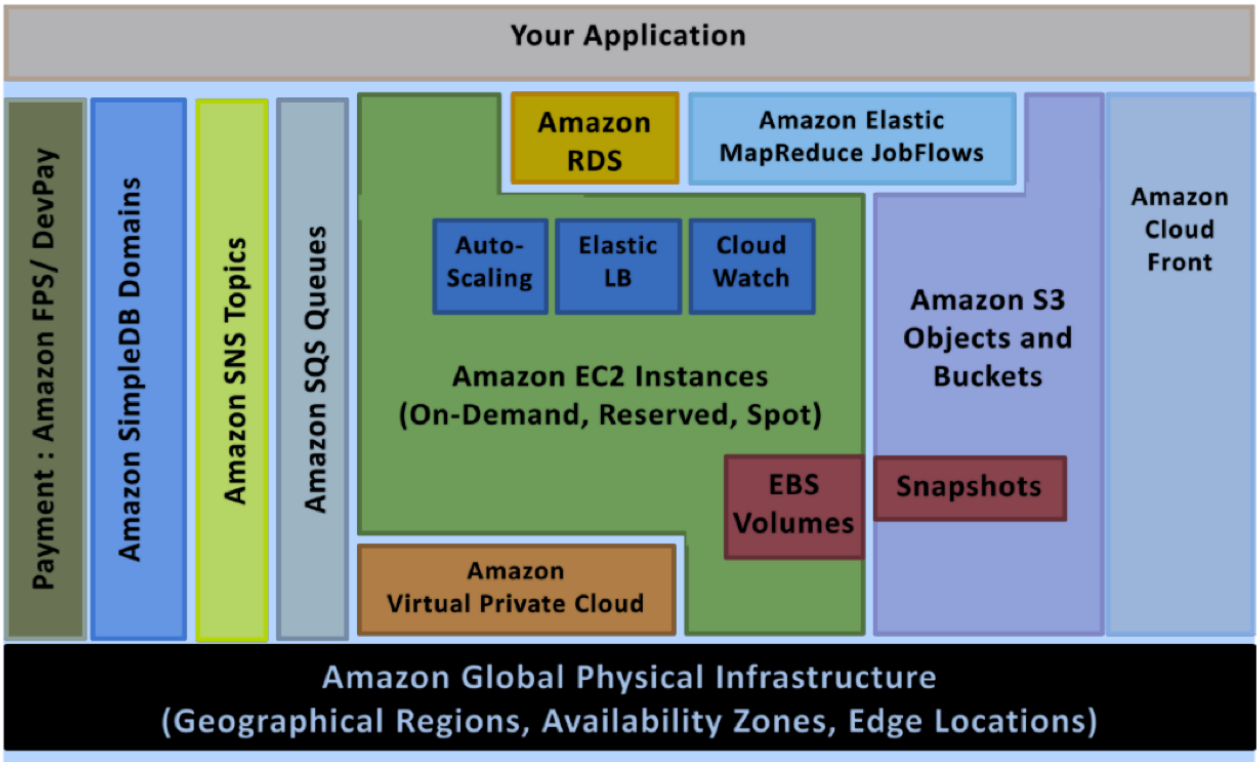
Private Cloud of Company XYZ with 3 business units, each with different security, SLA, governance and chargeback policies on shared infrastructure



Public Cloud Provider with 3 business customers, each with different security, SLA, governance and billing policies on shared infrastructure

3 Case Study

Amazon Web Services and the ECS2 Compute Cloud

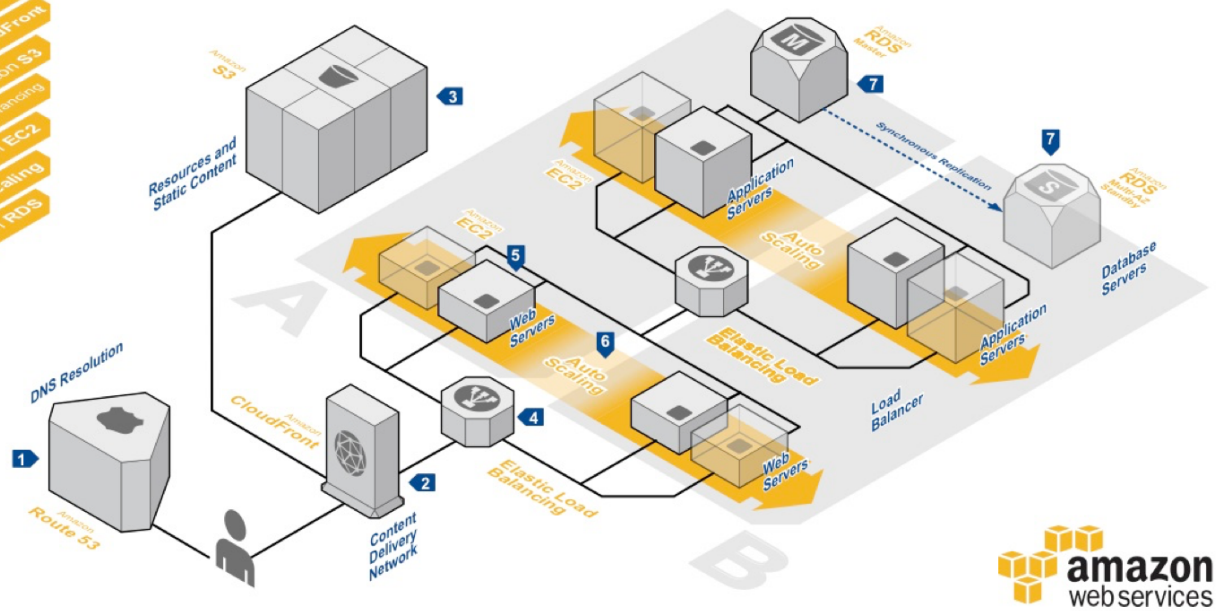

Your Application
Payment : Amazon FPS/ DevPay
Amazon SimpleDB Domains
Amazon SNS Topics
Amazon SQS Queues
Amazon RDS
Amazon Elastic MapReduce JobFlows
Auto-Scaling
Elastic LB
Cloud Watch
Amazon EC2 Instances (On-Demand, Reserved, Spot)
EBS Volumes
Amazon Virtual Private Cloud
Amazon S3 Objects and Buckets
Snapshots
Amazon Cloud Front
Amazon Global Physical Infrastructure
(Geographical Regions, Availability Zones, Edge Locations)

AWS Reference Architectures

- Amazon Route 53
- Amazon CloudFront
- Amazon S3
- Elastic Load Balancing
- Amazon EC2
- Auto Scaling
- Amazon RDS

WEB APPLICATION HOSTING

Highly available and scalable web hosting can be complex and expensive. Dense peak periods and wild swings in traffic patterns result in low utilization of expensive hardware. Amazon Web Services provides the reliable, scalable, secure, and high-performance infrastructure required for web applications while enabling an elastic, scale-out and scale-down infrastructure to match IT costs in real time as customer traffic fluctuates.



amazon web services

System Overview

1. The user's DNS requests are served by **Amazon Route 53**, a highly available Domain Name System (DNS) service. Network traffic is routed to infrastructure running in Amazon Web Services.

2. Static, streaming, and dynamic content is delivered by **Amazon CloudFront**, a global network of edge locations. Requests are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

3. Resources and static content used by the web application are stored on **Amazon Simple Storage Service (S3)**, a highly durable storage infrastructure designed for mission-critical and primary data storage.

4. HTTP requests are first handled by **Elastic Load Balancing**, which automatically distributes incoming application traffic among multiple **Amazon Elastic Compute Cloud (EC2)** instances across Availability Zones (AZs). It enables even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic.

5. Web servers and application servers are deployed on Amazon EC2 instances. Most organizations will select an **Amazon Machine Image (AMI)** and then customize it to their needs. This custom AMI will then become the starting point for future web development.

6. Web servers and application servers are deployed in an **Auto Scaling** group. Auto Scaling automatically adjusts your capacity up or down according to conditions you define. With Auto Scaling, you can ensure that the number of **Amazon EC2** instances you're using increases seamlessly during demand spikes to maintain performance and decreases automatically during demand to minimize costs.

7. To provide high availability, the relational database that contains application's data is hosted redundantly on a multi-AZ (multiple Availability Zones–zones A and B here) deployment of **Amazon Relational Database Service** (Amazon RDS).

AWS Reference Architectures

- Amazon EC2
- Amazon EC2 Spot
- Amazon EMR
- Amazon S3
- Amazon CloudFront
- Amazon RDS

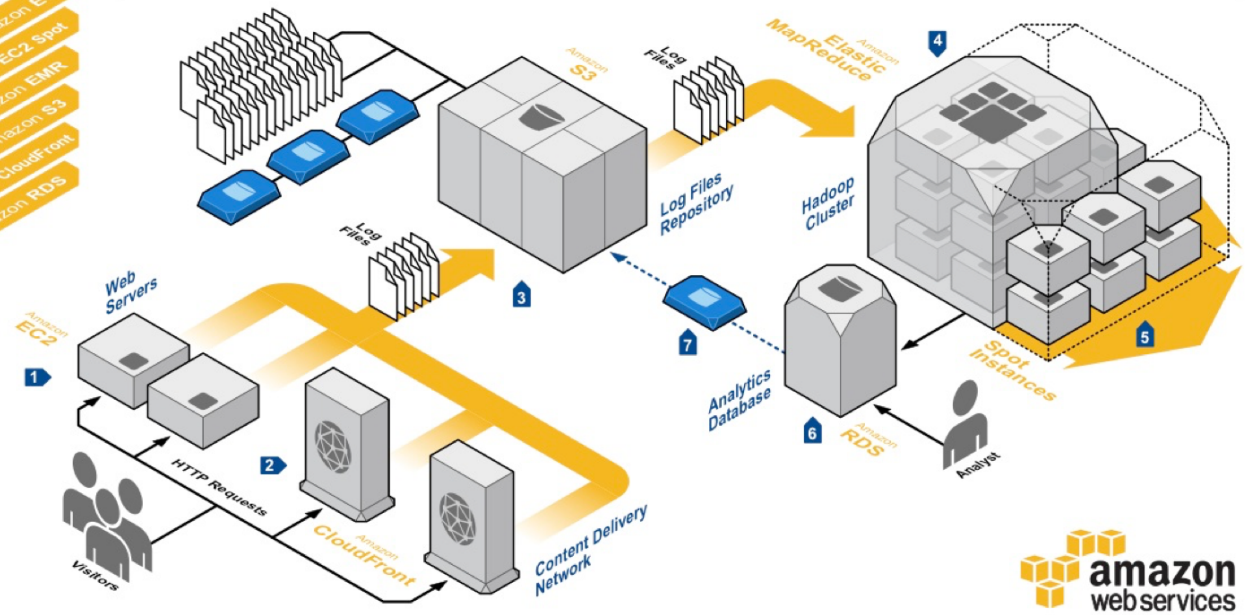
WEB LOG ANALYSIS

Amazon Web Services provides services and infrastructure to build reliable, fault-tolerant, and highly available web applications in the cloud. In production environments, these applications can generate huge amounts of log information.

This data can be an important source of knowledge for any company that is operating web applications. Analyzing logs can reveal information such as traffic patterns, user behavior, marketing profiles, etc.

However, as the web application grows and the number of visitors increases, storing and analyzing web logs becomes increasingly challenging.

This diagram shows how to use Amazon Web Services to build a scalable and reliable large-scale log analytics platform. The core component of this architecture is Amazon Elastic MapReduce, a web service that enables analysts to process large amounts of data easily and cost-effectively using a Hadoop hosted framework.



amazon web services

System Overview

1. The web front-end servers are running on **Amazon Elastic Compute Cloud** (Amazon EC2) instances.

2. **Amazon CloudFront** is a content delivery network that uses low latency and high data transfer speeds to distribute static files to customers. This service also generates valuable log information.

3. Log files are periodically uploaded to **Amazon Simple Storage Service** (Amazon S3), a highly available and reliable data store. Data is sent in parallel from multiple web servers or edge locations.

4. An **Amazon Elastic MapReduce** cluster processes the data set. **Amazon Elastic MapReduce** utilizes a hosted Hadoop framework, which processes the data in a parallel job flow.

5. When **Amazon EC2** has unused capacity, it offers EC2 instances at a reduced cost, called the *Spot Price*. This price fluctuates based on availability and demand. If your workload is flexible in terms of time of completion or required capacity, you can dynamically extend the capacity of your cluster using **Spot Instances** and significantly reduce the cost of running your job flows.

6. Data processing results are pushed back to a relational database using tools like *Apache Hive*. The database can be an **Amazon Relational Database Service** (Amazon RDS) instance. **Amazon RDS** makes it easy to set up, operate, and scale a relational database in the cloud.

7. Like many services, **Amazon RDS** instances are priced on a pay-as-you-go model. After analysis, the database can be backed-up into **Amazon S3** as a database snapshot, and then terminated. The database can then be recreated from the snapshot whenever needed.