

Introductory course to AWS Cloud and Core Services

Ioannis Chatzigiannakis

Sapienza University of Rome

Lecture 1

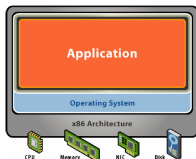
Virtualization

- Virtualization deals with “extending or replacing an existing interface so as to mimic the behavior of another system”
- Virtual system examples:
 - virtual private network,
 - virtual memory,
 - virtual machine,
 - ...



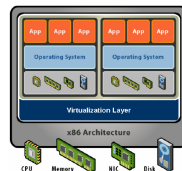
Starting Point: Physical System

- Physical Hardware
 - Processors, Memory, I/O devices, ...
 - Physical resources often underutilized
 - Periods that are over-utilized
- Software:
 - Tightly coupled to Hardware,
 - Single active OS,
 - OS controls Hardware



What is a Virtual Machine?

- Hardware-level Abstraction
 - Virtual Hardware: Processors, Memory, I/O devices, ...
 - Encapsulates all OS and application state.
- Virtualization Software:
 - Extra level of indirection decouples hardware and OS,
 - Multiplexes physical hardware across multiple “guest” VMs,
 - Strong isolation between VMs,
 - Manages physical resources, improves utilization.



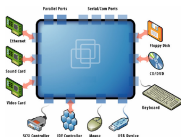
Virtual Machine Isolation

- Secure Multiplexing:
 - Run multiple VMs on single physical host,
 - Processor hardware isolates VMs.
- Strong Guarantees:
 - Software bugs, crashes, viruses within one VM cannot affect other VMs
- Performance Isolation:
 - Partition system resources,
 - Example: VirtualBox controls for reservation, limit, shares.



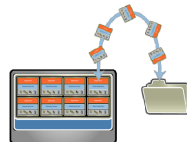
Virtual Machine Compatibility

- Hardware Independent:
 - Physical hardware hidden by virtualization layer,
 - Standard virtual hardware exposed to VM.
- Create Once, Run Anywhere:
 - No configuration issues,
 - Migrate VMs between hosts.
- Legacy Virtual Machines:
 - Run legacy OS on new platform.



Virtual Machine Encapsulation

- Entire VM in a file:
 - OS, applications, data;
 - Memory and device state.
- Snapshots and Clones:
 - Capture VM state on the fly and restore to point-in-time,
 - Rapid system provisioning, backup, remote mirroring.
- Easy Content Distribution:
 - Pre-configured apps, demos.
 - Virtual Appliances.



Common Uses

- Test and Development
 - Rapidly provision test and development servers.
 - Store libraries of pre-configured test machines.
- Business Continuity
 - Reduce cost and complexity by encapsulating entire systems into single files
 - Replicated and restored on demand into any target system.
- Enterprise Desktop
 - Secure unmanaged PCs without compromising end-user autonomy by layering a security policy in software around desktop virtual machines.



Common Uses

- Run legacy software on non-legacy hardware
- Run multiple operating systems on the same hardware
- Create a manageable upgrade path
- Manage outages (expected and unexpected) dynamically

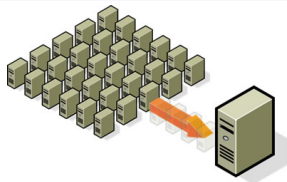


Non-virtualized Data Centers

- Too many servers for too little work
- High costs and infrastructure needs
 - Maintenance
 - Networking
 - Floor space
 - Cooling
 - Power
 - Disaster Recovery



Virtualized Data Centers



Reduce costs by consolidating services onto the fewest number of physical machines



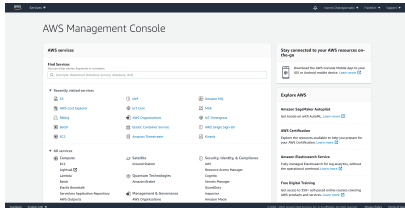
Dynamic Data Centers

- Virtualization helps us break the “one service per server” model
- Consolidate many services into a fewer number of machines when workload is low, reducing costs
- Conversely, as demand for a particular service increases, we can shift more virtual machines to run that service
- We can build a data center with fewer total resources, since resources are used as needed instead of being dedicated to single services

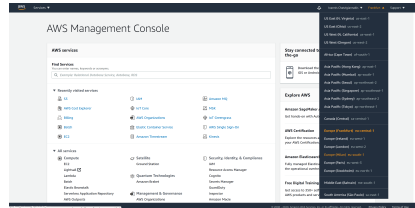
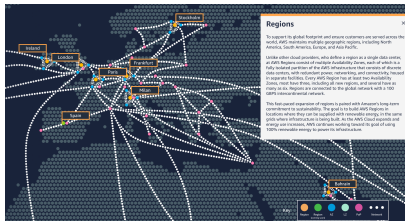


Function as a Service

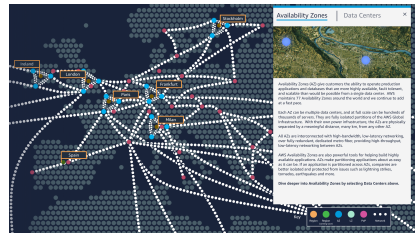




AWS Infrastructure



AWS Infrastructure



AWS Infrastructure



Introduction to AWS S3



Introduction to AWS S3

- S3 = Simple Storage Service
 - From 0 bytes to 5 Tbytes.
- Provides a secure, durable, highly-scalable storage space.
 - AWS secures content with encryption, ACL and bucket policies.
 - AWS guarantees 99.999999999% durability ($11 \times 9s$).
 - AWS guarantees 99.99% availability.
- We can access items stored:
 - Using the web.
 - Using the Web Console.
 - Using the Smartphone App.
 - From the Command line AWS tool.
 - Programmatically through the AWS S3 API.

S3 Basics

- Object-based storage.
 - Files = Objects.
 - Not suitable to install an operating system or host a database.
- Files/Objects are organized in Buckets.
- Bucket names must be unique – **S3 is a universal namespace.**
 - `http://sapienza2020adm.s3.amazonaws.com/`
 - When you create a new S3 bucket, AWS creates a new web address.
- Objects (Files) have the following properties:
 - Key: the name of the object.
 - Value: the actual contents.
 - Version ID: used by the versioning system.
 - Metadata: tags that we can attach to objects.
 - ACL: who can access the object.



S3 Storage Classes

- Free Tier – new AWS accounts
 - 5GB of S3 storage.
 - 20,000 GET – 2,000 PUT/COPY/POST/LIST
 - 15GB of Data Transfer Out each month for one year
- S3 Standard
 - \$0.0245 per GB
 - \$0.0054 per 1000 PUT/COPY/POST/LIST
 - \$0.00043 per 1000 GET/SELECT/all other requests.
- S3-IA Infrequent Access
 - \$0.0135 per GB – a minimum storage duration of 30 days.
 - \$0.01 per 1000 PUT/COPY/POST/LIST
 - \$0.001 per 1000 GET/SELECT/all other requests.
- S3 Glacier
 - \$0.0045 per GB – a minimum storage duration of 90 days.
 - \$0.06 per 1000 PUT/COPY/POST/LIST
 - \$0.00043 per 1000 GET/SELECT/all other requests.

