CIS 4517: Data Intensive and Cloud Computing

Cloud Computing service models and Amazon Cloud

Dr. Xubin He

Computer and Information Sciences

Temple University

Outline

- **Cloud computing service models**
- **Amazon Cloud: AWS**
  - AWS overview
  - Application examples
  - Using Amazon EC2 instances
- **Other cloud platforms**
  - Google Cloud and Microsoft Azure

Courtesy: Some slides are adopted and modified from C. Borcea.

Review from last class

- **Parallel and distributed systems:**
  - The fraction of the sequential part of a program limits the maximum performance speedup: Amdahl's Law

$$Execution\ time_{new} = Execution\ time_{old} \times \left((1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}} \right)$$

$$Speedup_{overall} = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

  - Long latency to remote memory: even a very small percentage of instructions involving remote memory access will have a big impact on the overall performance.
  - Tradeoff: Program needs sufficiently large units of work to run fast in parallel (i.e., large granularity), but not so large that there is not enough parallel work.

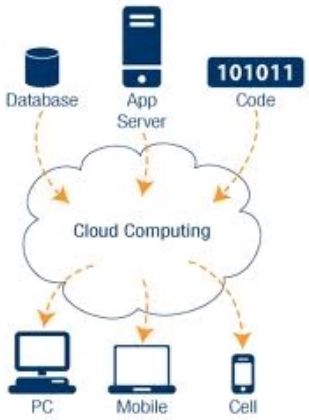
Load balancing

- **Load imbalance leads to some idle processors in the system**
  - Insufficient parallelism (during that phase)
  - Unequal size tasks

- **Program needs to balance load**
  - Sometimes can determine workload, divide up evenly, before starting: **static load balancing**
  - Sometimes workload changes dynamically, need to rebalance dynamically: **dynamic load balancing**

Cloud computing at a glance

- **Utility computing: our data and applications are hosted somewhere on the Internet ("in the cloud")**
  - Most services we access over the Internet are in the cloud (e.g., Google, Amazon, Yahoo)



Cloud infrastructure = Data centers with 100,000's servers

- Benefits:
  - Providers: economies of scale by having many users sharing the same infrastructure
  - Consumers: reduced cost and overhead

Cloud requirements (Yahoo!)

- **Multi-tenancy: Many apps co-existing on the same infrastructure**
- **Elasticity: Fast and graceful response to changing resource requirements**
- **Scalability: Scaling to growing data and apps**
- **Load and tenant balancing: absorbing load spikes, not to overload the hardware**
- **Availability: the cloud must be (almost) always on**
- **Security: No security breach into the cloud**
- **Metering: Monitoring cloud usage for resource provisioning and billing**
- **Simple APIs: Simplify deploying and tuning applications in the cloud**

Technical view

- **Computing resources**
  - Clusters of computers (massive parallelism)
  - Virtualization (sharing of resources)
  - Large scale storage facilities (access to data)

- **Network**
  - Web services

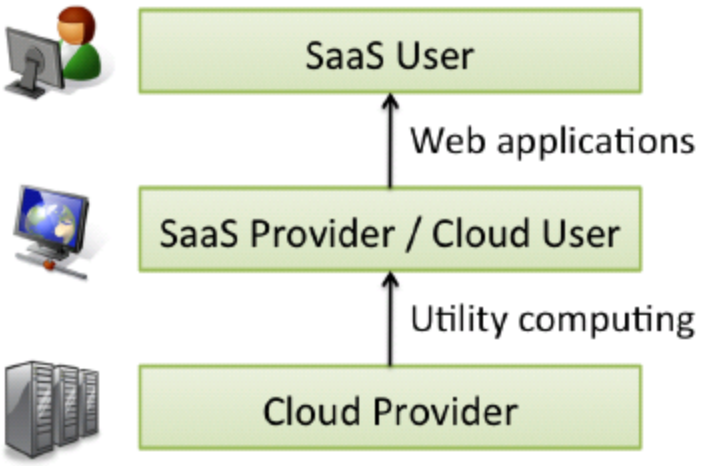
Cloud entities

Cloud providers provide various services to cloud customers.

- **Cloud providers: offer hardware & management tools**
  - provide system services (e.g., local OS + persistent storage + system software like compilers)
  - Can provide other (user-level) services (such as email)
- **Cloud customers**
  - Users: use the services offered by service providers

Types of services (X as a Service)

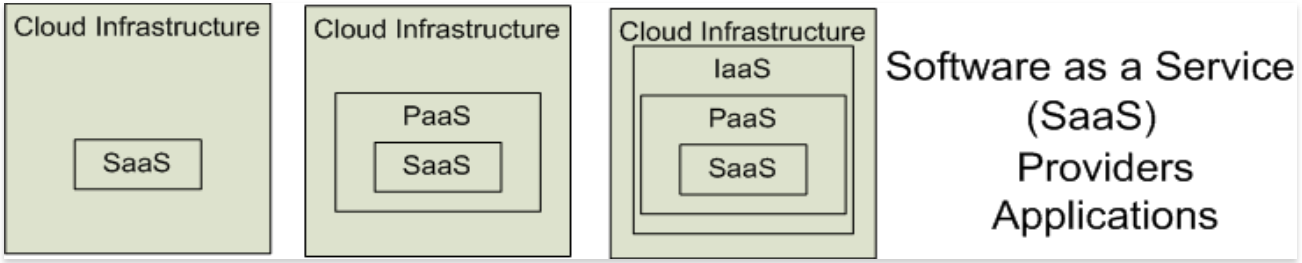
- **Infrastructure-as-a-Service (IaaS)**
  - Virtual servers with unique IP addresses and blocks of storage on demand (e.g., Amazon EC2, Google Compute Engine)
- **Platform-as-a-Service (PaaS)**
  - Set of software and development tools (API) hosted on the provider's servers (e.g., Windows Azure, Google AppEngine)
- **Software-as-a-Service (SaaS)**
  - The provider allows the customer only to use its applications (e.g., web-based email, web stores, etc)



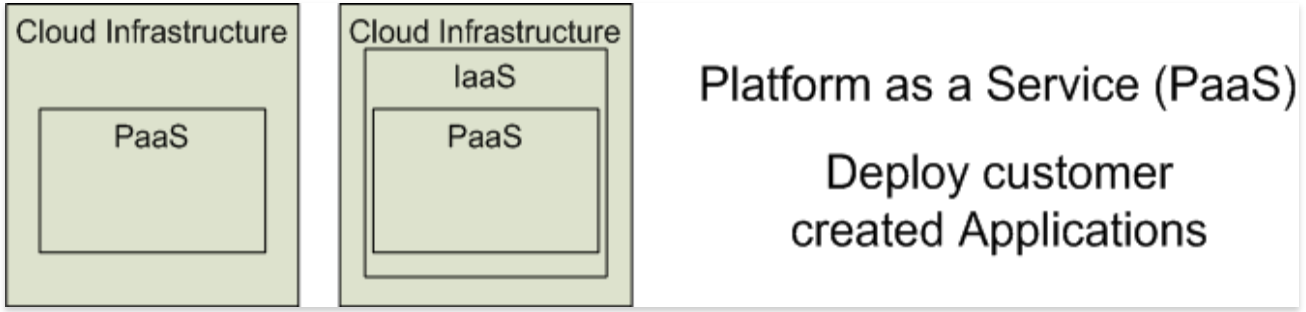
Cloud Service Models

Software as a Service (SaaS) | Platform as a Service (PaaS) | Infrastructure as a Service (IaaS)

SalesForce
CRM
LotusLive

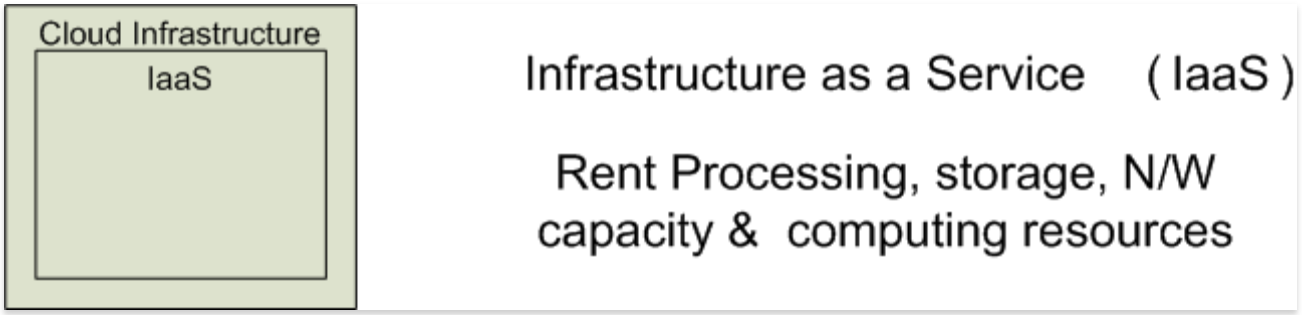


Windows Azure
The Future Made Familiar



amazon web services

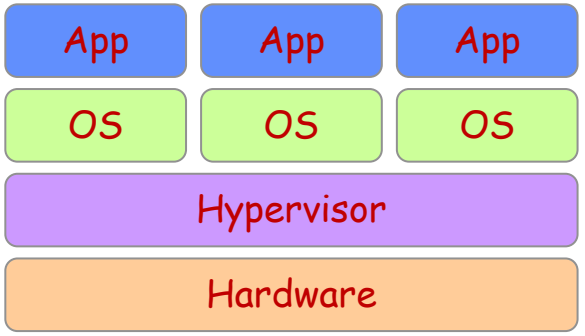
rackspace HOSTING



Effectively and Securely Using the Cloud Computing Paradigm by peter Mell, Tim Grance

Virtualization

- **Virtual workspaces:**
  - An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols,
  - Resource quota (e.g. CPU, memory share),
  - Software configuration (e.g. O/S, provided services).
- **Implement on Virtual Machines (VMs):**
  - Abstraction of a physical host machine,
  - Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs,
  - VMWare, Xen, KVM, etc.
- **Provide infrastructure API:**
  - Plug-ins to hardware/support structures



Virtualized Stack

New application opportunities with cloud

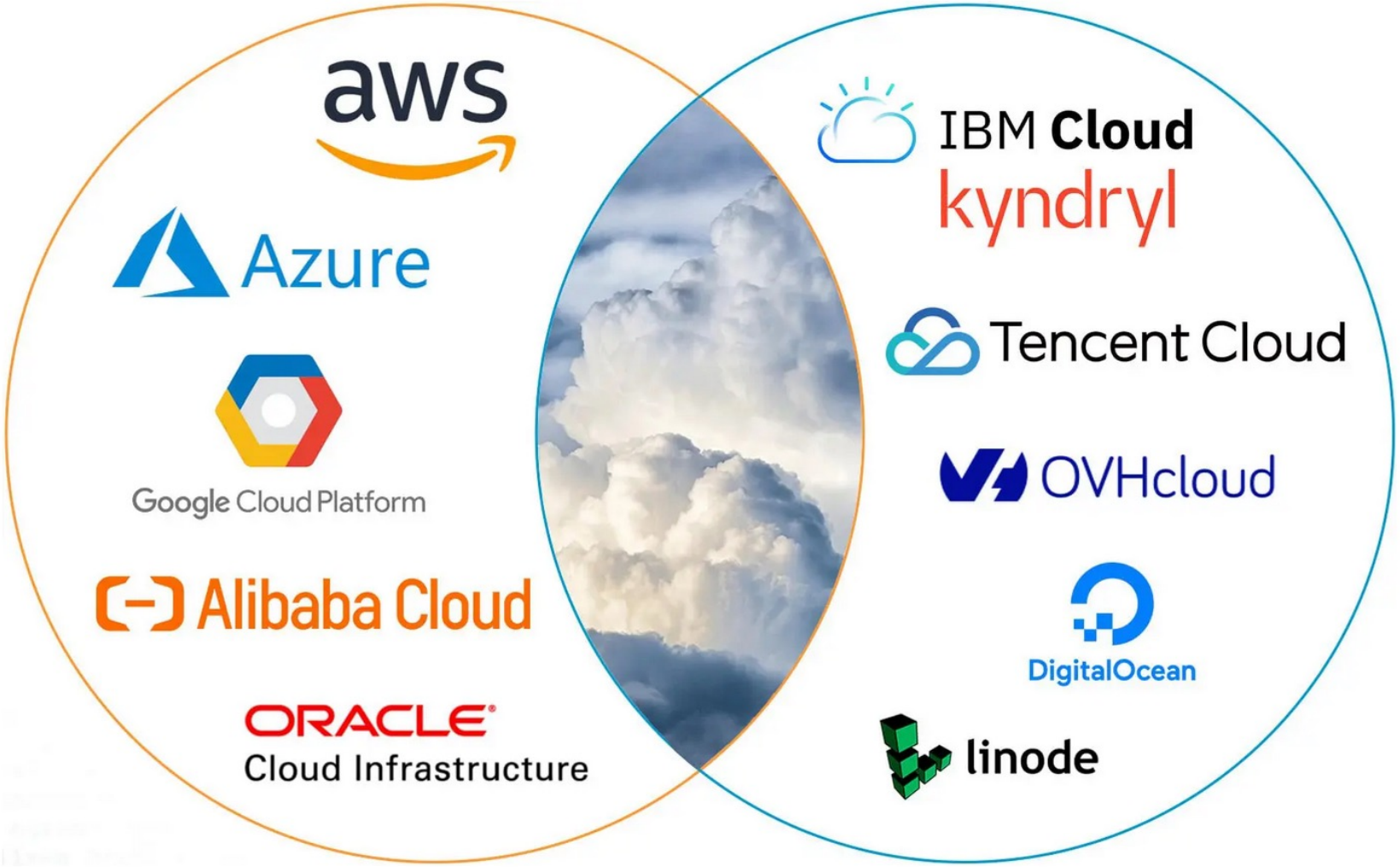
- **Parallel batch processing**
  - Get fast answers when processing large amounts of data
  - "Using hundreds of computers for a short time costs the same as using a few computers for a long time"
- **Mobile applications & services**
  - Store large data sets & perform complex computations in the cloud (e.g., scientific analysis, augmented reality)
  - Could maintain a virtual copy of your device in the cloud & potentially offload computations there
- **Extensions of desktop software**
  - Matlab, Mathematica, MS office

Business view

- **Shift in economic model: the relevance and weight of technology diminishes as it becomes commodity and standardized (electricity grid, phone, water supply)**

- **Providers: economies of scale by having many users sharing the same infrastructure**
  - Main reason why cloud computing happened when very large data centers have started to appear
- **Consumers: reduced cost and overhead**

Top cloud players



Source: https://dgtlinfra.com/top-10-cloud-service-providers-2022/

Provider's benefits

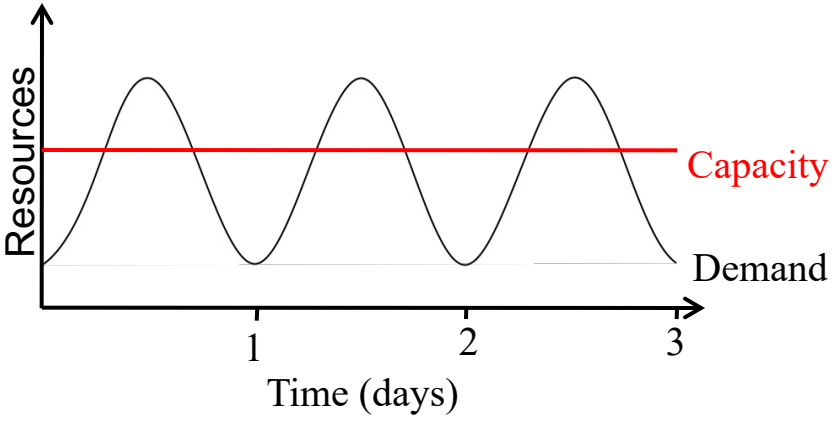
- Renting an IT infrastructure to many users
- Sharing of resources
- Reduction of costs through scale
- Centralized monitoring and maintenance
- Control over software evolution
- Control over service level agreements

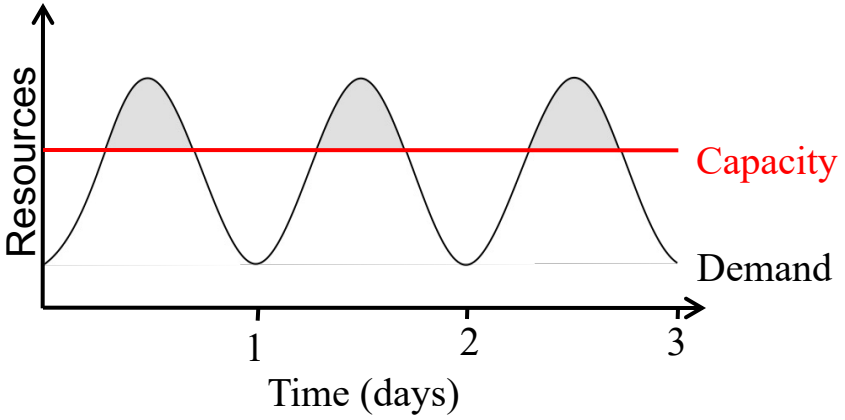
Consumer's benefits

- **Reduced capital expenditure for hardware, software, services**
- **Reduced operational expenses - Pay as you go (dynamic provisioning)**
- **Simple to achieve scalability and flexibility**
- **More predictable costs**

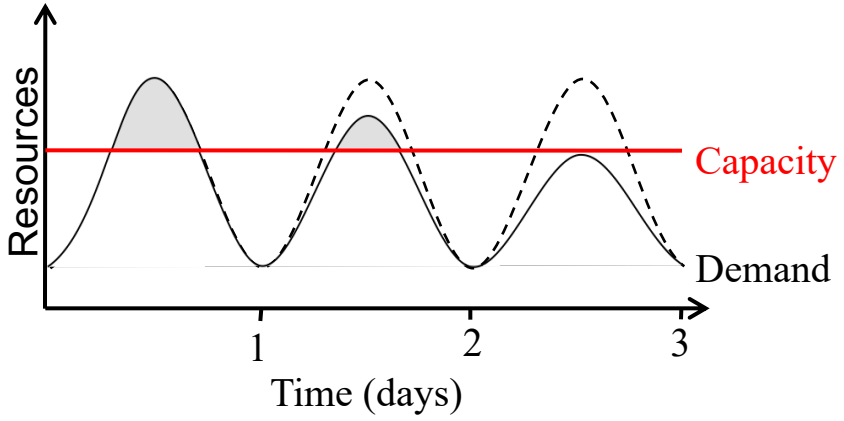
- **Complete a task in 1 hour using 100 machines vs 100 hours using 1 machine.**

Traditional data center: heavy penalty for under-provisioning





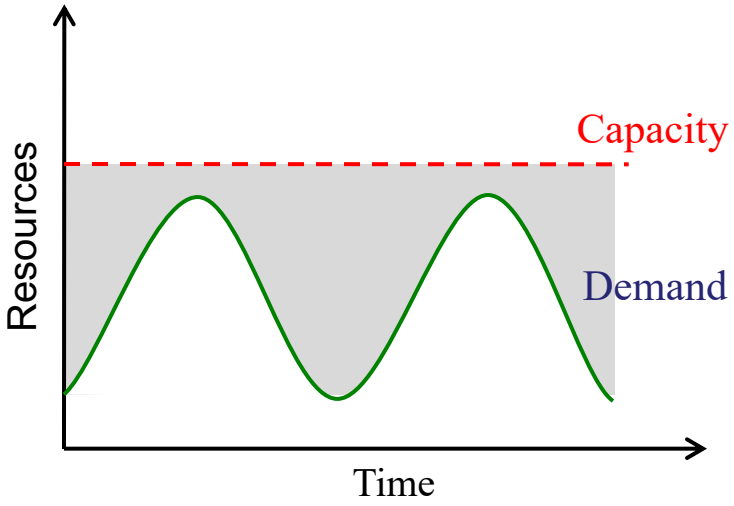
Lost revenue



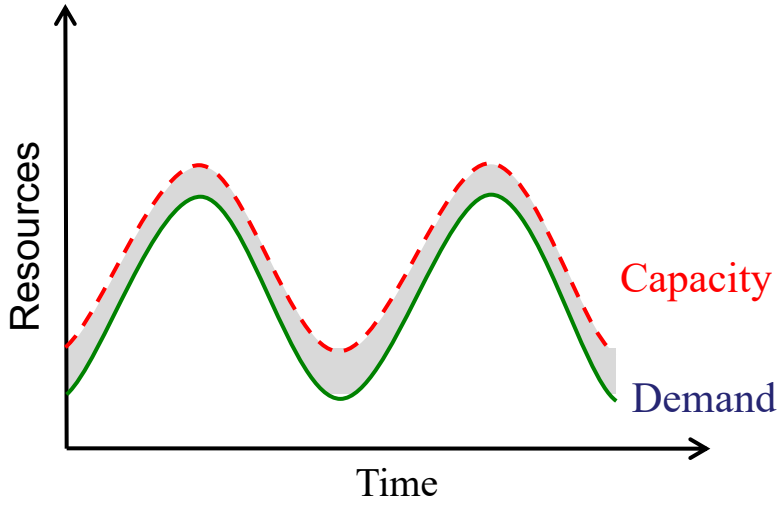
Lost users

- Over-provisioning?

Economics of cloud model



Static data center



Data center in the cloud

Unused resources

Drawbacks of the cloud

- Too much control on the provider side
- Privacy and security
- Legislation related to data management
- Data movement cost

IDC cloud report

- ***Cloud Going Mainstream: All Are Trying, Some Are Benefiting; Few Are Maximizing Value*, by R. Mahowald et al, September 2016.**

Academic clouds

- Emulab: https://www.emulab.net/
- PlanetLab: http://www.planet-lab.org/

- Chameleon Cloud: https://www.chameleoncloud.org/

- CloudLab: https://www.cloudlab.us/, build your own cloud on their hardware

Academic Cloud: Emulab
https://www.emulab.net/

- **A community resource open to researchers in academia and industry.**
- **Widely used by researchers everywhere today.**

- **Founded and owned by University of Utah (led by Late Prof. Jay Lepreau)**

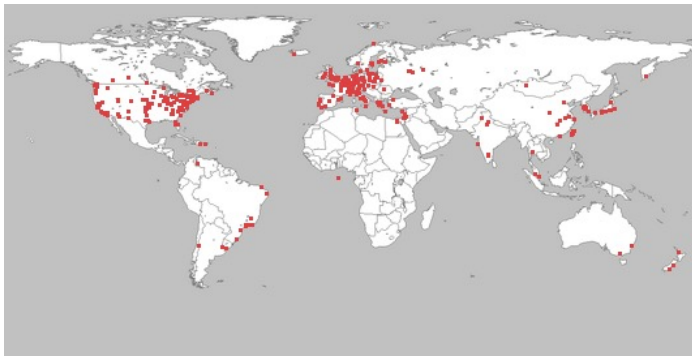
- **As a user, you can:**
  - Grab a set of machines for your experiment
  - You get root-level (sudo) access to these machines
  - You can specify a network topology for your cluster
  - You can emulate any topology

emulab
total network testbed

Slide source by IG: https://courses.engr.illinois.edu/cs425/fa2019/L2-3.FA19.ppt

PLANETLAB

An open platform for developing, deploying, and accessing planetary-scale services



All images © PlanetLab

- **A community resource open to researchers in academia and industry**
- **http://www.planet-lab.org/**
- **Currently, ~ 1077 nodes at ~500 sites across the world**
- **Founded at Princeton University (led by Prof. Larry Peterson), but owned in a federated manner by the sites**

- **Node: Dedicated server that runs components of PlanetLab services.**
- **Site: A location, e.g., UIUC, that hosts a number of nodes.**
- **Sliver: Virtual division of each node. Currently, uses VMs, but it could also other technology. Needed for timesharing across users.**
- **Slice: A spatial cut-up of the PL nodes. Per user. A slice is a way of giving each user (Unix-shell like) access to a subset of PL machines, selected by the user. A slice consists of multiple slivers, one at each component node.**
- **Thus, PlanetLab allows you to run real world-wide experiments.**
- **Many services have been deployed atop it, used by millions (not just researchers): Application-level DNS services, Monitoring services, CoralCDN, etc.**
- **PlanetLab is basis for NSF GENI https://www.geni.net/**

Slide source by IG: https://courses.engr.illinois.edu/cs425/fa2019/L2-3.FA19.ppt

Cloud platform example: AWS

AWS Overview

Application example

Using Amazon EC2 instances

AWS Services

Compute
- EC2
- EC2 Container Service
- Lightsail
- Elastic Beanstalk
- Lambda
- Batch

Storage
- S3
- EFS
- Glacier
- Storage Gateway

Database
- RDS
- DynamoDB
- ElastiCache
- Redshift

Networking & Content Delivery
- VPC
- CloudFront
- Direct Connect
- Route 53

Migration
- DMS
- Server Migration
- Snowball

Developer Tools
- CodeCommit
- CodeBuild
- CodeDeploy
- CodePipeline

Management Tools
- CloudWatch
- CloudFormation
- CloudTrail
- Config
- OpsWorks
- Service Catalog
- Trusted Advisor
- Managed Services
- Application Discovery Service

Security, Identity & Compliance
- IAM
- Inspector
- Certificate Manager
- Directory Service
- WAF & Shield
- Compliance Reports

Analytics
- Athena
- EMR
- CloudSearch
- Elasticsearch Service
- Kinesis
- Data Pipeline
- QuickSight

Artificial Intelligence
- Lex
- Polly
- Rekognition
- Machine Learning

Internet Of Things
- AWS IoT

Game Development
- GameLift

Mobile Services
- Mobile Hub
- Cognito
- Device Farm
- Mobile Analytics
- Pinpoint

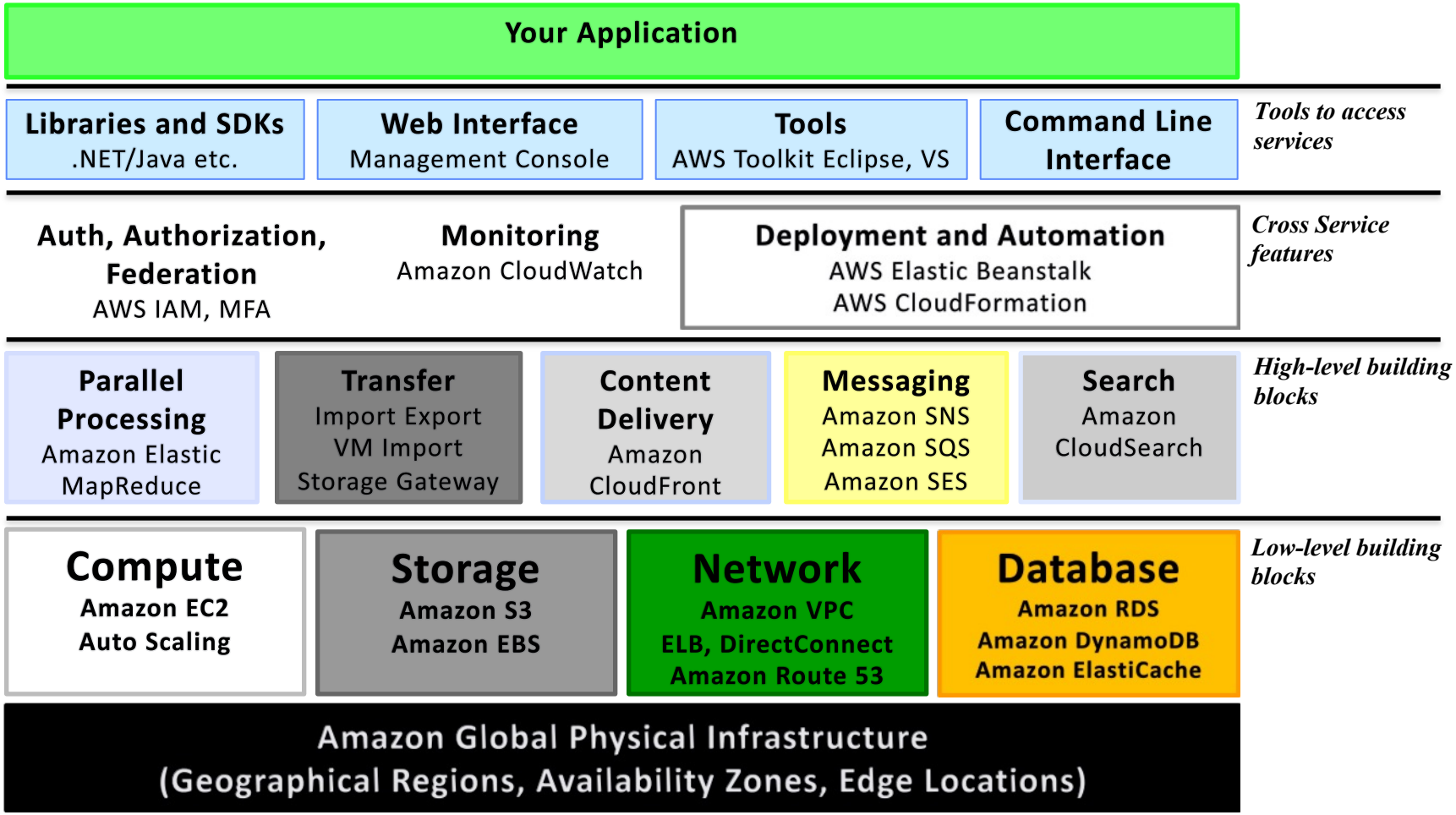
Application Services
- Step Functions
- SWF
- API Gateway
- Elastic Transcoder

Messaging
- SQS
- SNS
- SES

Business Productivity
- WorkDocs
- WorkMail

Desktop & App Streaming
- WorkSpaces
- AppStream 2.0

The AWS Cloud For Enterprises



Note: This slide is adopted from J. Varia@Amazon

AWS service: Compute and monitoring

- **Compute**
  - *Elastic Compute Cloud (EC2):* launch on-demand virtual machines (instances)
  - *Elastic MapReduce*: automatically starts Hadoop implementation of MapReduce for parallel applications
    - » Amazon handles cluster management
  - *Auto Scaling:* seamlessly increase/decrease number of EC2 instances function of load
    - » Done based on metrics reported by CloudWatch
- **Monitoring**
  - *CloudWatch*: monitor cloud resources such as CPU cycles, disk access, network traffic

AWS Service: Storage

- *Simple Storage Service (S3):* provide persistent object storage
  - » Independent of EC2 instances
  - » EC2 instances need to "download" data from S3 in order to access it (cannot issue read/write to S3)
- Amazon Glacier: low-cost storage service that provides secure and durable storage for data archiving and backup
  - » Advantage over S3: offload the administrative burdens of operating and scaling storage + cost
  - » Disadvantage: slower than S3
- *Elastic Block Store (EBS):* provide block level storage volumes (virtual disk, i.e., disk-like) to EC2 instances
  - » Persistent even after instances are terminated
  - » Instances have to mount EBSs (EFS)

AWS service: Database

- DynamoDB: non-relational database service, fully-managed, high performance, easy to set up, operate, and scale
- *Relational Database Service (RDS):* full-featured MySQL database
- Amazon ElastiCache: a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud
  - » Load data in memory and access it there instead of going to disk
  - » http://memcached.org

AWS service: Messaging

– *Simple Queue Service (SQS):* reliable & scalable mechanism for asynchronous communication
  » Asynchronous communication simplifies fault-tolerance
  » Can be used for communication between EC2 instances or between machines outside the cloud and EC2 instances

– *Simple Notification System (SNS):* event-driven system that sends "push" notifications (from the cloud) to multiple applications
  » Publish-subscribe mechanism

– *Simple Email Service (SES):* yet another email system

AWS Service: Networking

- **Networking**
  - *Route 53:* DNS service that provides automatic availability & scalability
  - *Virtual Private Cloud:* allows integration of company machines with EC2 instances in one network (using IPSec VPN)
  - *Elastic Load Balancing:* automatically distributes incoming traffic to multiple EC2 instances
  - *AWS Direct Connect:* establish a dedicated connection from your premise to AWS (1GB/10GB fiber)
- **Web**
  - *CloudFront:* CDN (content distribution network) service
    - » Like all CDNs uses caches at the edge of the network
    - » Works for both static content and streaming
  - *Alexa Web Information Service (AWIS):* provides web analytics

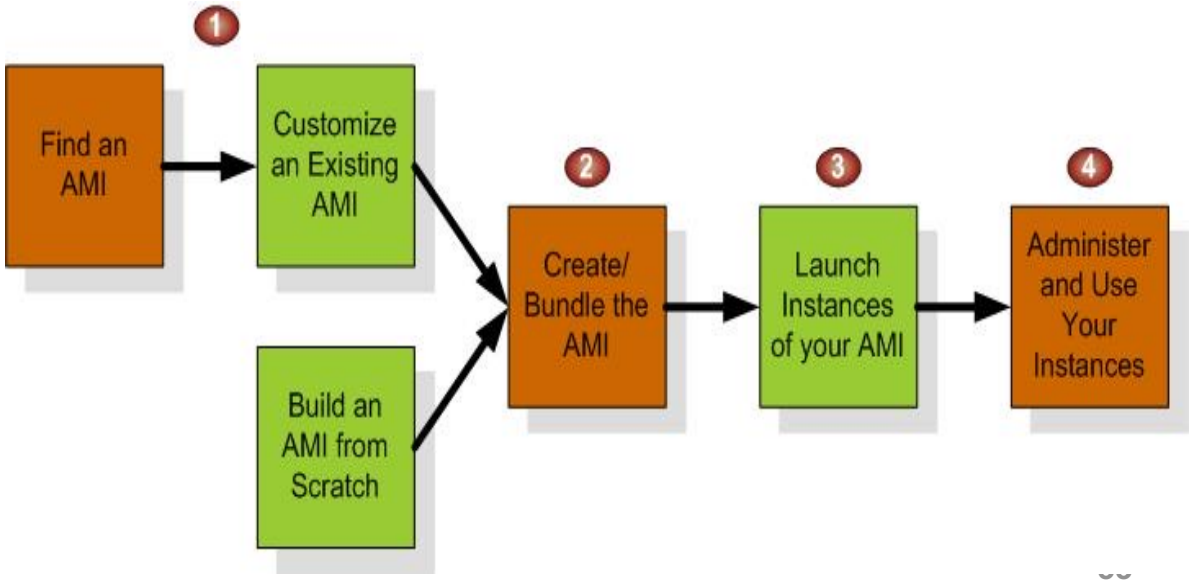
EC2 instances

- **Instances: Virtual machines that run in the EC2 environment**
  - Each instance is like a "physical" machine that has its own CPU, memory, network interface, and disk space (volatile – data is lost when the instance is terminated)
  - *Xen* used for virtualization
- **AMI (Amazon Machine Image): Encrypted file that captures a complete snapshot of an EC2 instance at a point in time, including its software, configuration, and data**
  - Images are stored in S3 and serve as boot disks for instances
  - Linux/Solaris and Windows images publicly available
  - Users can create AMI from scratch: start from any public AMI, install & customize the software needed, and then store it as private AMI to use later on

EC2 environment

- **Provides instance management and configuration services**
  - Launch and terminate instances
  - Control the instance properties (e.g., type of instance, AMI) through a simple web interface
  - Set network access permissions for instances

Instance creation:



Auto Scaling

- **Auto Scaling helps maintain application availability and allows users to scale an Amazon EC2 capacity up or down automatically according to defined conditions.**

- **To ensure that you are running your desired number of Amazon EC2 instances.**

- **Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during low demands to reduce costs.**

Simple Storage Service (S3)

- **Concepts**
  - Objects and keys: an object (with an associated key) is stored in S3
  - Buckets: where the data is stored ("unlimited storage")
- **A bucket is a container for objects and describes location, logging, accounting, and access control. A bucket can hold any number of objects, which are files of up to 5TB. A bucket has a name that must be globally unique.**
- **Fundamental operations corresponding to HTTP actions:**
  - http://**bucket**.s3.amazonaws.com/**object**
  - POST a new object or update an existing object.
  - GET an existing object from a bucket.
  - DELETE an object from the bucket
  - LIST keys present in a bucket, with a filter.
- **A bucket has a flat directory structure (despite the appearance given by the interactive web interface.)**

Simple Storage Service (S3)

- **Resources are identified by URIs**
  - Example: http://cis4517.s3-us-west-2.amazonaws.com/CloudChronicles.pdf
  - cis4517 is the bucket
  - CloudChronicles.pdf is the object
- **Has an access control mechanism to allow/deny access**
- **Limitations**
  - Not possible to modify a small portion of a file
    - » Need to rewrite the whole object again with the modification
  - Changes take time to propagate

Elastic Block Storage (EBS)

- **EBS are raw unformatted persistent virtual disks for EC2 instances**
  - Size: up to 20 TB (1TB=2^{40} bytes)
  - Each user can use up to 5000 EBS volumes
- **An EBS can be attached to only one instance at a time**
  - EBS can be used as boot partition for instances: fast startup time
- **An instance can mount multiple EBSs**
- **Volumes are replicated for availability**
- **Snapshot feature**
  - Users can create incremental snapshots to S3
    - » Good for sharing & instantiating new volumes
    - » Extra availability

Elastic Block Store

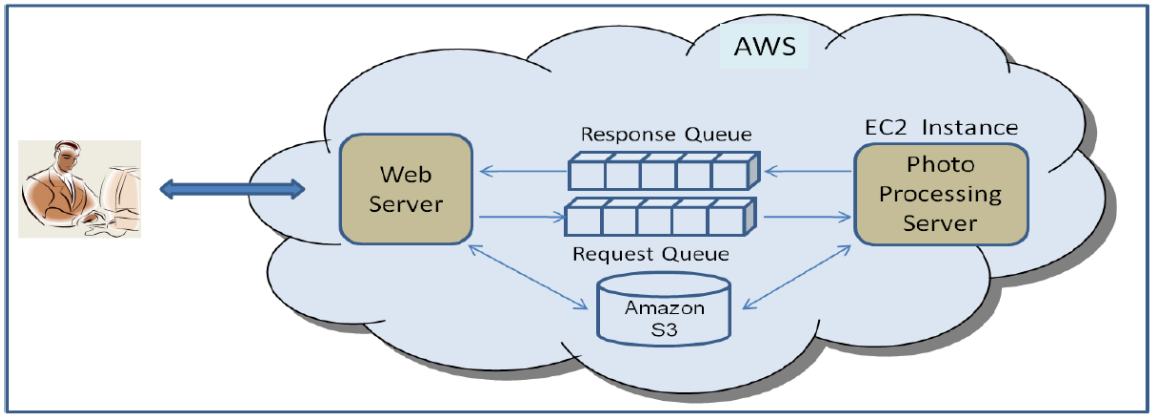
- **An EBS volume is a virtual disk of a fixed size with a block read/write interface. It can be mounted as a filesystem on a running EC2 instance where it can be updated incrementally. Unlike an instance store, an EBS volume is persistent.**
- **(Compare to an S3 object, which is essentially a file that must be accessed in its entirety.)**
- **Fundamental operations:**
  - CREATE a new volume (1GB-1TB)
  - COPY a volume from an existing EBS volume or S3 object.
  - MOUNT on one instance at a time.
  - SNAPSHOT current state to an S3 object.

Outline

- **Cloud computing service models**
- **Amazon Cloud: AWS**
  - AWS overview
  - Application examples
  - Using Amazon EC2 instances

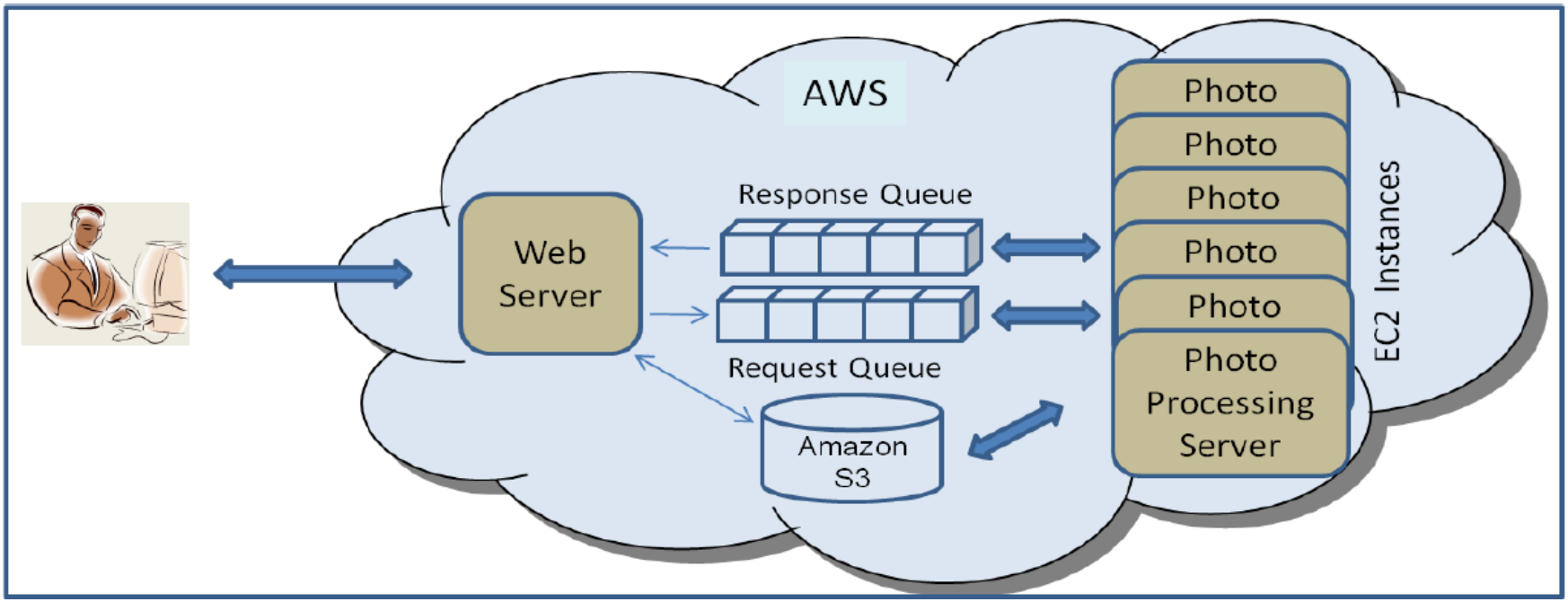
Online photo processing service (1)

- **Users submit Photos and specify operations they want over their photos:**
  - Red eye detection, Cropping, Re-coloring, etc



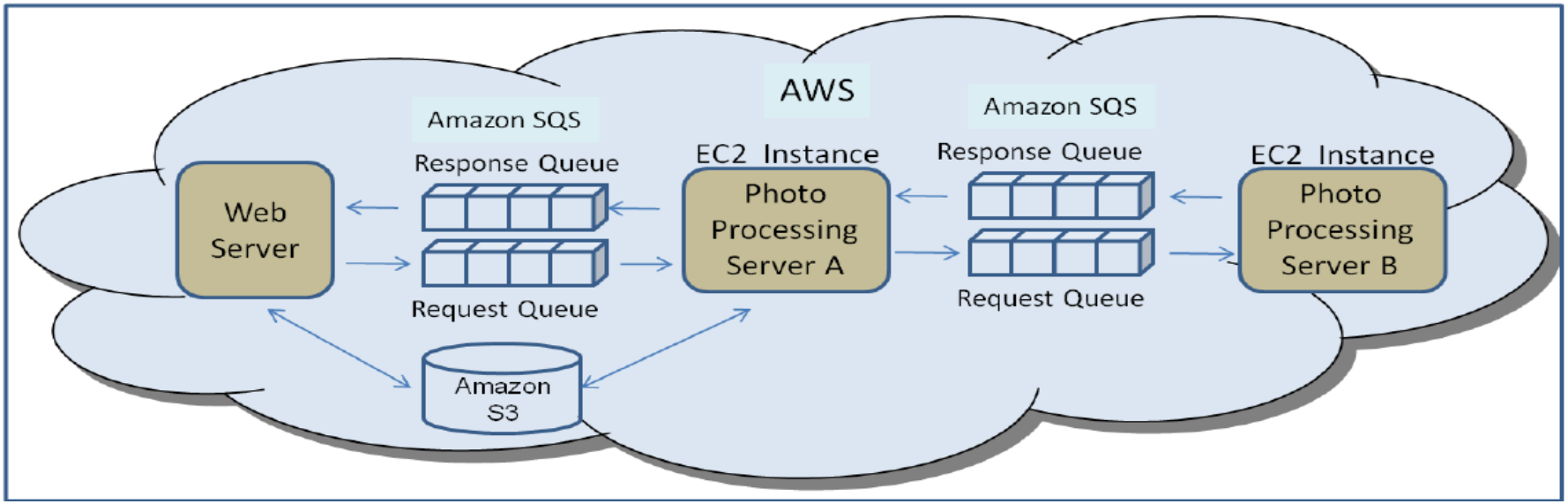
- Requests are put on Request Queue and photos are saved in S3 storage area
- Photo Processing Server gets the request, retrieves photos from S3 storage, performs processing, and sends result using Response Queue

Online photo processing service (2)



- If more than one instance is needed, user can initiate more instances
- Message queue can be used by multiple instances
- Message retrieved by one instance is locked, such that other instances cannot see it

Online photo processing service (3)



- **Pipeline processing: if an operation takes more time, user can create another instance to handle the time consuming jobs**
  - Suppose server B performs the time consuming jobs
  - Server A and B can communicate using SQS

Case study

- **The New York Times used AWS to create PDF files of its whole archive**
  - 100 Amazon EC2 instances running Hadoop application
  - Processed 4TB of raw TIFF image data (stored in S3) into 11 million finished PDFs
  - Running time: 24 hours
  - Cost: $240 (not including bandwidth)

New York Times report

Outline

- **Cloud computing service models**
- **Amazon Cloud: AWS**
  - AWS overview
  - Application examples
  - Using Amazon EC2 instances

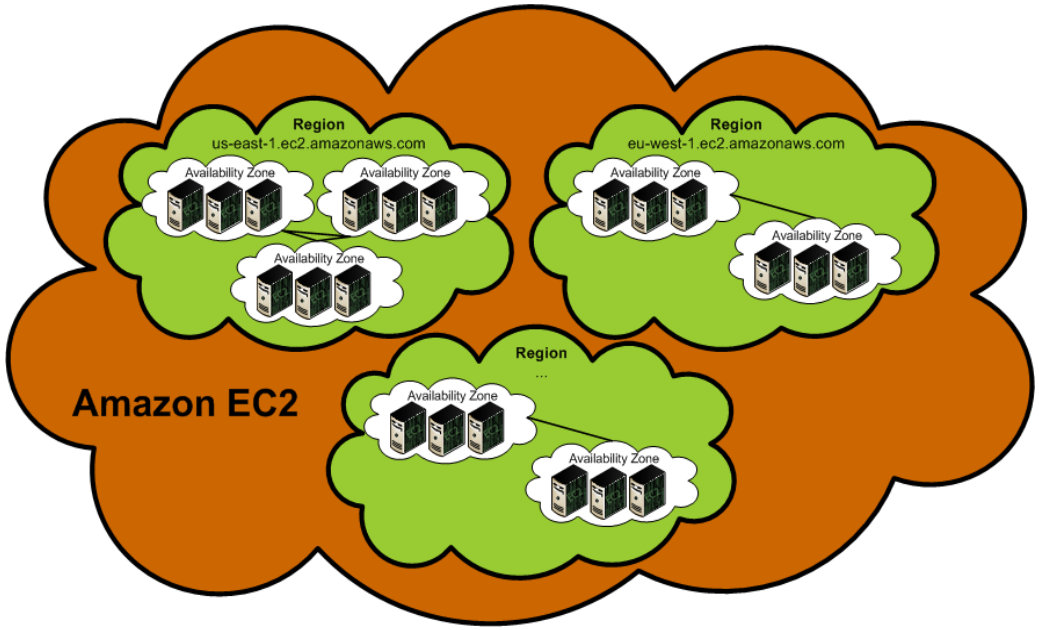
Getting Started with Amazon EC2

- **Step 1: Sign up for Amazon EC2**
- **Step 2: Create a key pair**
- **Step 3: Launch an Amazon EC2 instance**
- **Step 4: Connect to the instance**
- **Step 5: Customize the instance**
- **Step 6: Terminate instance and delete the volume created**

Terminology

- **Instance: One running virtual machine.**
- **Instance Type: hardware configuration: cores, memory, disk etc.**
- **Amazon Machine Image (AMI): Description of an instance. It provides the information required to launch an instance.**
- **Key Pair: Credentials used to access VM from command line.**
- **Region: Geographic location, price, laws, network locality.**
- **Availability Zone: Subdivision of region that is fault-independent.**
- **EBS: Elastic Block Store: provides persistent block level storage volumes for use with Amazon EC2 instances in the AWS Cloud.**

Regions and availability zones



- **Regions are located in separate geographic areas (US: Virginia & California, Ireland, Singapore, etc.)**
  - Each Region is completely isolated
  - Failure independence and stability
- **Availability Zones are distinct locations within a Region**
  - Isolated, but connected through low-latency links
  - Failure resilience

AWS Global Infrastructure

- **The AWS Cloud spans 102 Availability Zones within 32 geographic regions around the world, with announced plans for 12 more Availability Zones and 4 more AWS Regions (as of September 2, 2023).**

Source: https://aws.amazon.com/about-aws/global-infrastructure/

Instance types

- **Some typical instance types:**
  - General purpose: mac, T2, T3, M3, M4, M5x, M6x, M7x…
  - Compute optimized: C7x, C6g, C4…
  - Memory optimized: R7x, R6x, R5x, X1, R4, R3
  - Accelerated computing instances (eg. GPUs): P5, P4, P3, P2, G2…
  - Storage optimized: I4x, I3, I2, D2
  - HPC optimized: hpc7x, hpc6x

https://aws.amazon.com/ec2/instance-types/

EC2 Pricing Model

- **Free Usage Tier: Free EC2 instances, including 750 hours of Linux and Windows t2.micro instances (t3.micro for the regions in which t2.micro is unavailable), each month for one year.**
- **On-Demand Instances**
  - Start and stop instances whenever you like, costs are rounded up to the nearest hour. No contract. (Worst price)
- **Reserved Instances**
  - Pay up front for one/three years in advance. (Lower price)
- **Spot Instances**
  - Specify the price you are willing to pay, and instances get started and stopped without any warning as the marked changes. (Bid)
- **Dedicated Hosts**
  - A Dedicated Host is a physical EC2 server dedicated for your use.

Current pricing: http://aws.amazon.com/ec2/pricing/

Demo: launch an EC2 instance

- **A video is available for you to access in course canvas (under Resources tab).**

- **Once you are done, remember to stop your instance (but don't terminate it). If you stop your instance, later you can re-start it. However, if you terminate your instance, your EC2 is gone and all your work on this EC2 is lost.**

Next

- **More Cloud platform practices:**
  - Google Cloud
  - Microsoft Azure

- **Create your own AWS account if you haven't done so and play with it. It's free!**

- **Homework: Launching & Configuring an EC2 Instance (details please refer to the course Canvas).**