Cloud Intro / AWS Overview

Franky Romero and Mark Zhao

announcements

- Sign up on Google Form for paper presentations
  - Need group for next Monday

- OH for paper projects (by appt?)

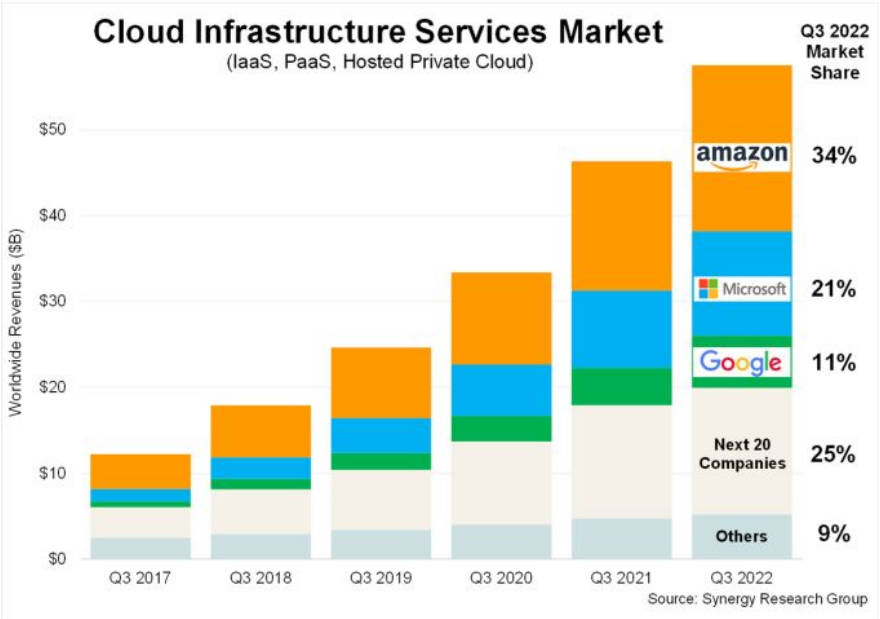
Review: What is the cloud?

“On-demand delivery of [IT resources] … **via the internet** with **pay-as-you-go pricing**.”

- **On-demand**
  - Stop guessing capacity, right-size for demand
  - Agility to adopt new resources/advancements
- **Via the internet**
  - Avoid direct datacenter costs
  - Global deployment
- **Pay-as-you-go pricing**
  - Fixed expense -> variable expense
  - Economies of scale

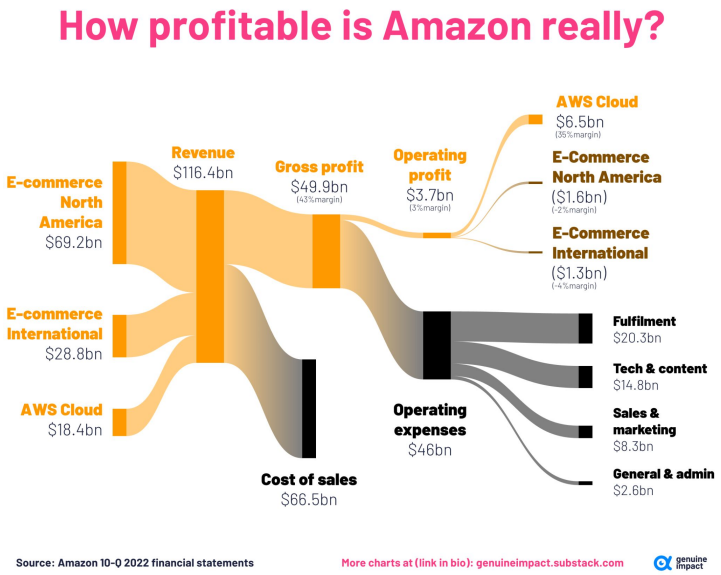
Cloud Service Providers

Who’s used a cloud service before? What are some similarities and differences across providers?



What is AWS?

- Provides suite of cloud **services** (200+)



Question: What is a service?

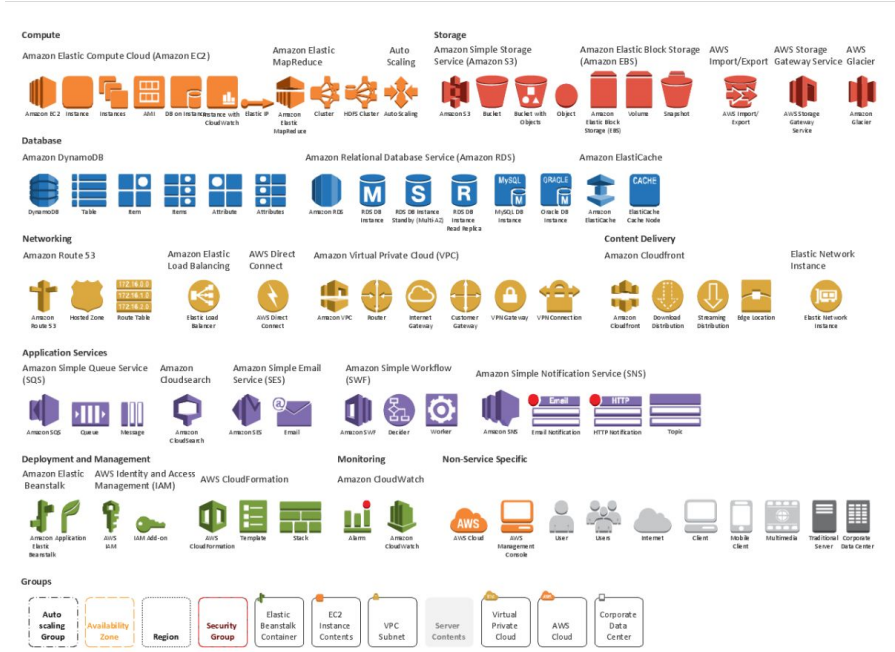
Review: Cloud Service Types

Cloud computing models: IaaS, PaaS, SaaS

- Infrastructure as a Service
  - Basic infrastructure building blocks (VMs, networking, storage)
  - AWS Example: EC2
- Platform as a Service
  - Automatically manages underlying infrastructure (capacity, load balancing, auto-scaling)
  - AWS Example: Elastic Beanstalk
- Software as a Service
  - Complete product managed by the service provider
  - AWS Example: Rekognition

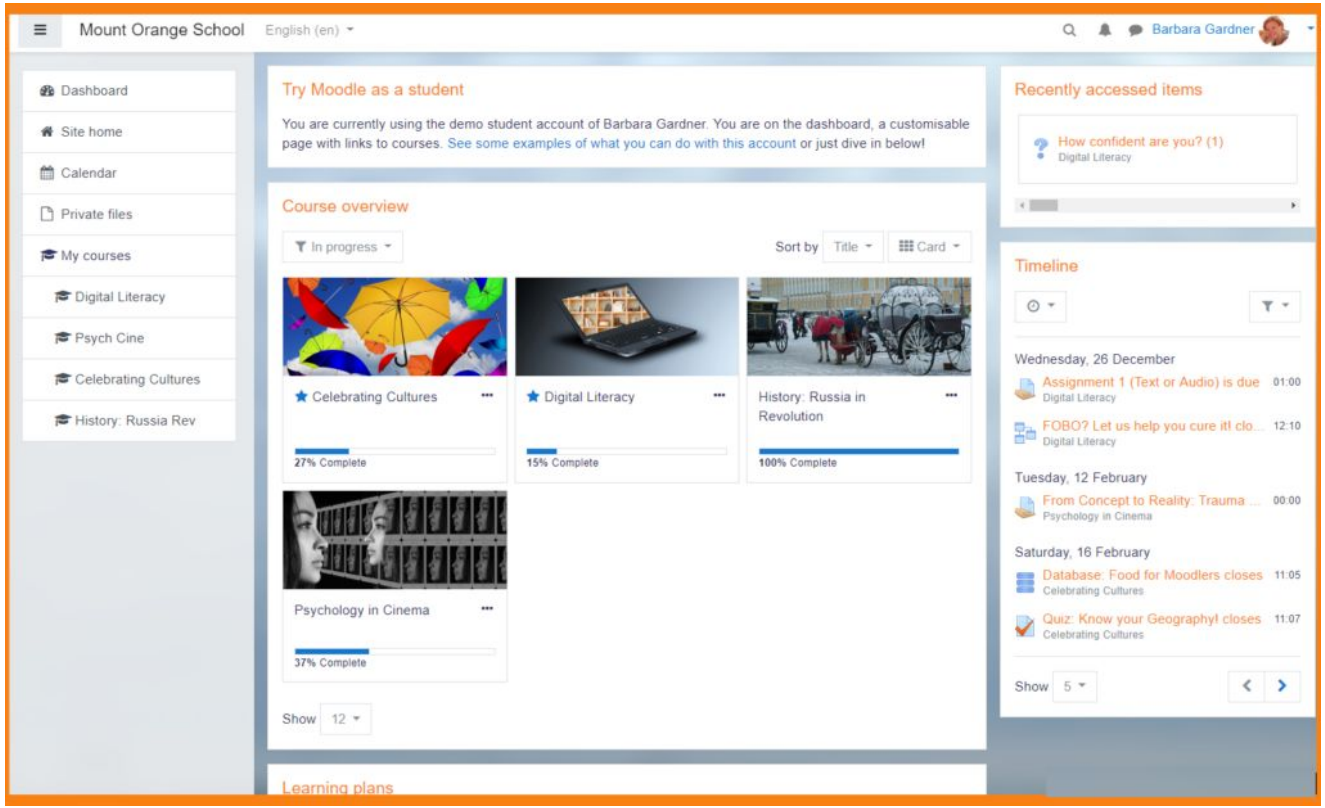
Question: What does Serverless fall under?

AWS Services



Question: What do you think the most important AWS services are?

Let’s build Moodle

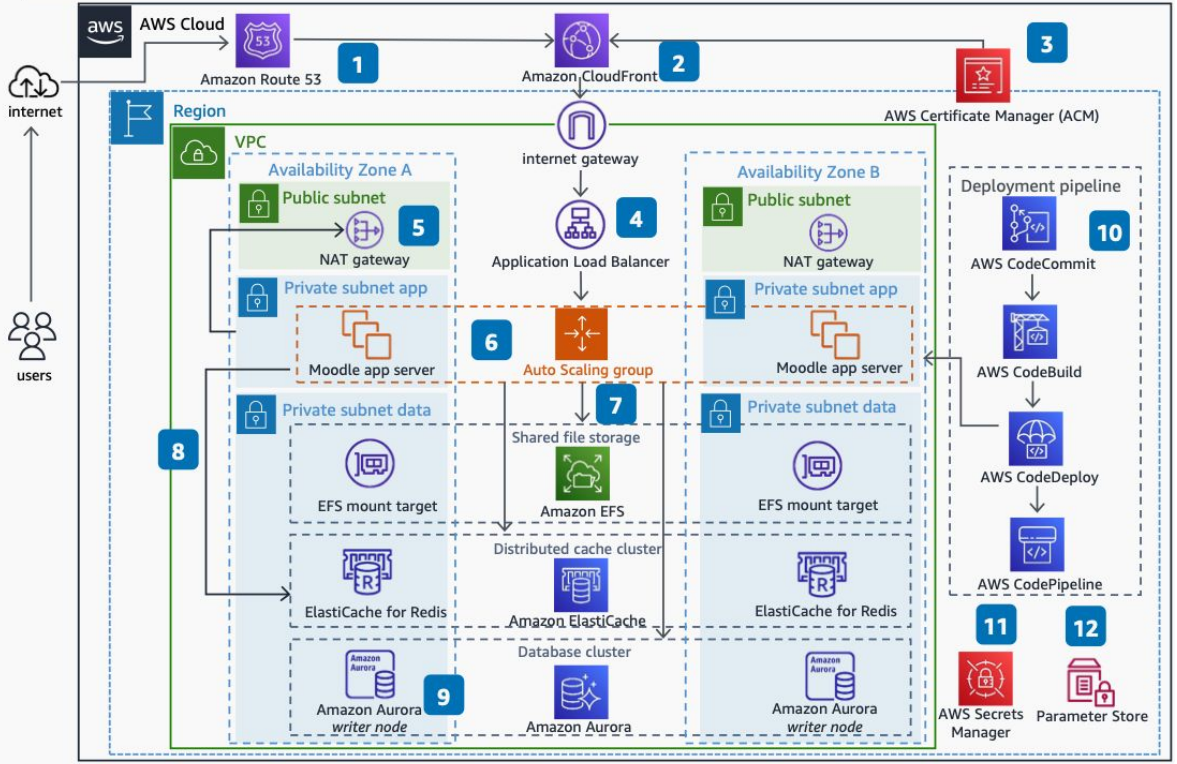


Let’s build Moodle with AWS

- User types “myschool.edu/myclass” and logs into course page to watch latest lecture
  - DNS resolution: Amazon Route 53
  - Load balancer: Amazon Application Load Balancer
  - Secure communication (SSL): AWS Certificate Manager
  - Web Server: Amazon EC2
  - Content Distribution Network: Amazon CloudFront
- User uploads a discussion post
  - Storage system: Amazon EFS or S3
  - Session/application cache: Amazon ElastiCache
  - Database: Amazon Aurora or DynamoDB
- Developer updates website
  - CodeCommit->CodeBuild->CodeDeploy->CodePipeline

Moodle Reference Architecture

Moodle is a learning platform designed to provide educators, administrators and learners with a single robust, secure, and integrated system to create personalized learning environments. Moodle learning management system (LMS) on AWS can be deployed using architecture and can scale up on demand. Separate the application and data layers for elasticity and security.

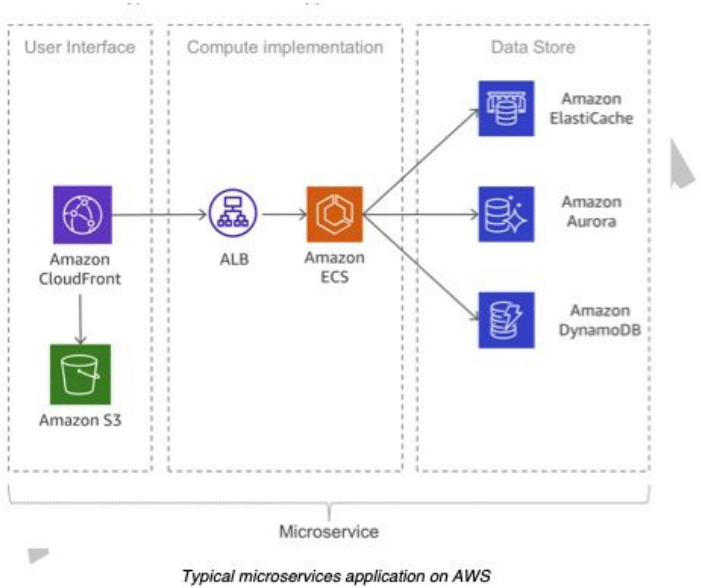


https://aws.amazon.com/architecture/reference-architecture-diagrams

Microservices Architecture

- What is a monolithic vs microservice architecture?
  - Microservice: Application composed of small, independently-scalable services that communicate over well-defined APIs

Question: What are the benefits of microservice architectures?

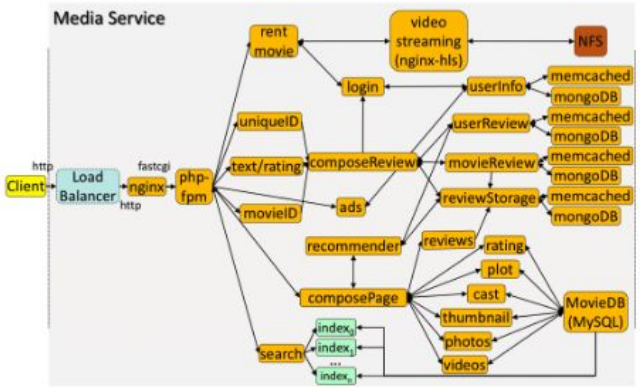


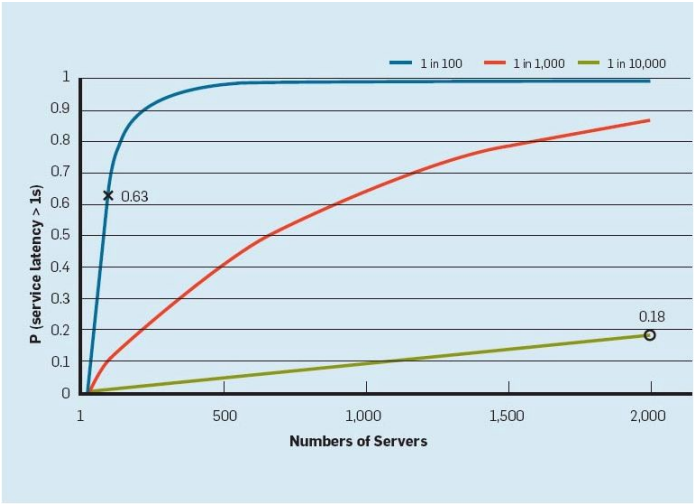
Typical microservices application on AWS

Quality-of-service in microservice architectures

What are some drawbacks?

- Each service has multiple dependencies (high fan-out)
- Why do we care about tail latency?





Quality of Service Metrics

- How do we measure the “success” of a service?
- SLOs:
  - Latency
  - Uptime (9s)
  - Recovery point/recovery time
  - …
- …

How do we build a “Well-Architected” system on AWS?

Centered around six pillars

- Operational excellence
- Security
- Reliability
- Performance efficiency
- Cost optimization
- Sustainability

Operational Excellence

- Perform operations as code
  - commit/review config files
- Make frequent, small, reversible changes
  - small, reviewable commits
- Refine operations procedures frequently
- Anticipate failure
  - “storms” or “game days”, pre-mortems
- Learn from all operational failures
  - post-mortems - any interesting ones?

Security

- Implement a strong identity foundation
  - principle of least privilege
- Enable traceability
  - monitor/alerting
- Apply security at all layers
- Automate security best practices
- Protect data in transit and at rest
- Keep people away from data
- Prepare for security events

Reliability

- Automatically recover from failure
- Test recovery procedures
- Scale horizontality to increase aggregate workload availability
- Stop guessing capacity
- Manage change in automation

Questions:

- What are some other design patterns to help with service reliability?

Performance Efficiency

- Democratize advanced technologies
- Go global in minutes
  - Why do we need to distribute workloads across global regions?
- Use serverless architectures
  - Q: When is serverless *not* more efficient/performant?
- Experiment more often
- Consider mechanical sympathy

Cost Optimization

- Implement cloud financial management
- Adopt a consumption model
- Measure overall efficiency
- Stop spending money on undifferentiated heavy lifting
- Analyze and attribute expenditure

Sustainability

- Understand your impact
- Establish sustainability goals
- Maximize utilization
  - Energy-proportionality?
- Anticipate and adopt new, more efficient hardware and software offerings
  - Embodied vs Operational carbon footprint?
- Use managed services
  - Benefits of PUE?
- Reduce the downstream impact of your cloud workloads

What are some other methods to improve sustainability of cloud workloads?

Discussion Questions

- What are some surprising services offered by AWS? Were there any that you think should/shouldn't be offered?
- What are some pros/cons of building systems on top of IaaS vs PaaS vs SaaS solutions?
  - e.g., deploying an ML app on top of SageMaker vs EC2 vs Lamba?
  - How does cost factor into this?
- What are some of the pros/cons of using *all* of a provider's services versus building some of it yourself? How do you decide this?
- From the well-architected documentation, we see that logging and monitoring are important across all facets of one's system. What sorts of logging should we be doing, and what are the challenges with providing infrastructure for this?