Main areas

* Cloud concepts
  + What is the AWS cloud and its basic infrastructure
  + Describe basic AWS cloud architectural principles
  + Understand the AWS cloud value proposition
* Security
  + Describe basic security/compliance aspects of the AWS platform
  + Shared responsibility model
* Billing / Pricing
  + Define the billing, account management and pricing models
* Technology (services)
  + Key services and their common use cases
  + Identify sources of documentation or tech assistance
  + Describe basic characteristics of deploying and operating in the AWS cloud

Cloud computing – on-demand delivery of IT resources and applications via the internet

# Structure

AWS is divided in regions. Each region is a separate geographic area and has a number of Availability Zones (AV), which in turn have 1+ data centers.

* Regions
  + Separate geographic regions (good for data location from compliance standpoint)
    - Isolated form one another
  + Made up of 2+ AVs
  + Services might differ per region (older ones usually get first dibs on new services)
* Availability Zones (AVs)
  + Made up of 1+ data centers
  + Low latency communication between AVs within one region
* Edge locations
  + Nodes for a Global Delivery Network (CDN)
    - Amazon CloudFront
  + Do not provide EC2 instances, but are rather places to store static data for caching purposes (e.g. to have a video load faster for the local users)

Fault tolerance – a system can remain operational even if some of its components fail

Elasticity – the ability to scale computing resources up or down easily

# Shared responsibility model

Provider – security **OF** the cloud

* Buildings, employees, hardware (machines, networking components, cables, cooling)
* Patching the infrastructure and services (so, the software that enables the cloud to run. The one that enables the creation and functioning of a virtual machine, not the software on the virtual machine itself)

Customer – security **IN** the cloud

* Access to the cloud resources (training, permissions of own employees)
* Data security and encryption (in transit and at rest)
* OS, Network and Firewall configuration (and patching) – if included. E.g. EC2 is IaaS and so the guest OS is responsibility of the customer, but S3 is PaaS and so the underlying OS and networking is responsibility of Amazon
* All code deployed on an instance

# Economics of the cloud

Capitalized Expenditure (CapEx) – the upfront costs of building the data center.

Operating Expenditure (OpEx) – the consistent costs of running and operating the data center.

Traditional model – High initial CapEx (as well as for every upgrade). Consistent OpEx, independent on demand. Paying for unused capacity and still likely to end up unable to fully satisfy a spike in demand. Largely guess reliant, monthly costs map to your predictions.

Cloud model – No initial CapEx. Pay as you Go - OpEx dependent on demand (also lower than the traditional model as the OpEx is just the Amazon bill, and they are making large economies of scale on all their maintenance costs such as electricity, cooling, employees, etc..). Paying only for used capacity and able to quickly scale up or down to satisfy changes in demand (elasticity). Not at all guess reliant, costs map to actual demand.

Predicting and managing AWS costs

AWS TCO (Total Cost of Ownership) Calculator – gives a three year overview of predicted costs for infrastructure on-premises vs cloud. Calculates server, storage, networking and IT labor costs.

AWS Simple Monthly Calculator

AWS Cost Explorer

* Breakdowns by service, cost tag (that you’ve made)
* Predictions for the next three months (if current trends remain?)
* Recommendations for cost optimizations
* Accessible via API

AWS Organizations

* Allow you to manage multiple AWS accounts under a single master account
* Allows you to receive Consolidated Billing for all accounts (one total bill)
* Used to centralize logging and security standards across accounts (think data access)

# Architecture on AWS

Well Architected Framework – collection of best practices across five key pillars for how to best create systems that create business value on AWS

* Operational Excellence – running and monitoring systems
* Security – protecting information and business assets
* Reliability – infrastructure to recover from disruptions
  + Fault tolerance (able to support the failure of components of the infrastructure)
  + High Availability (entire solution working as it should despite any issues that might occur)
  + AWS Disaster Recovery Approaches (in order of simplicity)
    - Backup and Restore (only backups on AWS)
    - Pilot Light (minimal resources are setup in AWS to support a DR event)
    - Warm Standby (non-scaled, up-and-running copy of the systems. Can shift to them and scale them up in the event of a DR)
    - Multi-Site (basically splitting evenly between local and AWS)
* Performance Efficiency – using resources efficiently
* Cost Optimization – minimal costs for the desired value

Supporting AWS Infrastructure

* Basic Support – provided for all
  + Access to Trusted Advisor (7 Core Checks)
  + 24/7 access to Customer Support (ticket system, will be bottom of the queue though), Documentation, Forums and Whitepapers
  + Access to Personal Health Dashboard
  + No monthly cost
* Developer Support
  + All of the Basic Support
  + Business hours access to Support Engineers
  + Limited to 1 Primary Contact
  + 29$ per month (at least)
* Business Support
  + All of Dev Support
  + Full set of Trusted Advisor checks
  + 24/7 Phone, Email and Chat access to Support Engineers
  + Unlimited Contacts
  + 100$ per month (at least)
* Enterprise Support
  + All of Business Support
  + Designated Technical Account Manager (TAM)
  + Concierge Support Team
  + 15,000$ per month (at least)

AWS interfaces

* AWS Management Console (the online GUI) – ease of use
* Command Line Interface (CLI) – can automate, language agnostic
* Software Development Kits (SDKs) – same as above, but per language

Elastic IP address – refers to your account and you can map it to a certain machine. Can be used to build fault tolerance – if the machine in one AV fails, you can quickly remap the IP to refer to a machine in a different AV. Only IPv4 supported. Takes the place of the public IPv4 address (cannot use both at once (either the public, which is per instance, or the elastic one which is per account and can be quickly remapped). However, can have more than one elastic IP address per instance).

# Services

EC2 (ECC – Elastic Compute Cloud) –

S3 (SSS – Simple Storage Service) –

Amazon CloudFront –

DynamoDB –

ElasticBeanstalk –

Lambda –

Storage

Ephemeral (instance store) – temporary block-level store. Connected to a EC2 instance (practically a part of it) and located on the same server as the instance.

* Free with an EC2 instance
* Data is lost when the instance is terminated/stopped (but not when rebooted) – basically like the drive of a physical machine

S3 (Simple Storage Service) – unlimited bucket size, but limited individual object size to 5TB

* Priced for storage used. No cost on transfers in, but some on transfers out
* Not a file system (unlike, say, Dropbox) – cannot map a drive and make folders and such. It’s just a bucket with a bunch of files stacked there, which you can access individually
* 99.99% uptime SLA
* Two types of S3 storage
  + Granular storage type selection – within the same S3 bucket one can pick and choose which file should have which type of storage
  + Standard – 99.999999999 (9 times) % durability
    - First 50TB $0.023 / GB
    - So durable because everything gets copied multiple times in the same AV, and also in other AVs. Can sustain a failure of 2 AVs and still have the file
  + Reduced Redundancy Storage – only 99.99% durability
    - First 1TB $0.024 / GB
    - Copied at less places, **but** can set up notifications that X file is currently unavailable