AWS Intro, EC2, and Lambda: Core Concepts

Briefing Document: AWS Introduction, EC2, and Lambda

This briefing document summarizes the key concepts and information presented in the provided excerpts from "11 - AWS Intro.pdf" and "12 - EC2 & Lambda.pdf". It covers the fundamentals of Amazon Web Services (AWS), its core service categories, the shared responsibility model for security, its global infrastructure, and provides a deeper dive into two foundational compute services: EC2 (Elastic Cloud Compute) and Lambda.

Source 1: "11 - AWS Intro.pdf"

Main Themes:

- **Introduction to AWS:** AWS is presented as a leading cloud platform offering over 200 services globally through a network of regions and availability zones. A key characteristic is its "pay-as-you-use cost model," which is theoretically more cost-effective than traditional data center rentals.
- **History and Growth:** Launched in 2006 with S3 and EC2, AWS has seen significant expansion in its service offerings, reaching over 200 services by the present day. The platform actively spurred adoption through competitions and has continuously innovated across various domains like operations, development, and analytics.
- **Cloud Service Models (IaaS, PaaS, SaaS):** The document outlines the three primary cloud service models:
  - **IaaS (Infrastructure as a Service):** Provides basic building blocks for IT infrastructure.
  - **PaaS (Platform as a Service):** Removes the need for infrastructure management, allowing users to focus on application deployment.
  - **SaaS (Software as a Service):** Delivers fully managed software applications by a vendor.
- **The Shared Responsibility Model:** This is a crucial concept for understanding security in AWS. It clearly divides responsibilities between AWS ("Security OF the cloud") and the client ("Security IN the cloud").
  - **AWS Responsibilities:** Include the security of the physical infrastructure (data centers, networking, power, HVAC), the hypervisor, host operating systems for managed services, and the maintenance of the underlying infrastructure and server software for its managed services. As stated, AWS is responsible for the "*Security of physical infrastructure (infra) and network*".
  - **Client Responsibilities:** Encompass control of data and content (classification, encryption, sharing, data-handling policies), access management and IAM (configuration of users, roles, policies, and enforcing the "Principle of Least Privilege"), management of self-hosted applications and their OSs, network security within their VPC, and handling compliance and governance. Clients are responsible for "*Control of Data/Content*".

- **AWS Global Infrastructure:** The infrastructure is organized into:
  - **Regions:** Distinct geographical areas (e.g., us-east-1).
  - **Availability Zones (AZs):** Multiple isolated data centers within each region. These are "*roughly equiv to isolated data centers*".
  - **Edge Locations:** Used for content delivery networks (CDNs) and caching to bring content closer to users.
- **Key AWS Service Categories:** The document provides an overview of various service categories with examples:
  - **Compute Services:** Including VM-based (EC2), container-based (ECS, ECR, EKS, Fargate), and serverless (Lambda).
  - **Storage Services:** Such as object storage (S3), file systems (EFS), block storage (EBS), caching (Amazon File Cache), and backup (AWS Backup). Amazon S3 is described as "*Object storage in buckets; highly scalable; different storage classes*".
  - **Database Services:** Covering relational (RDS, Aurora), key-value (DynamoDB), in-memory (MemoryDB, ElastiCache), document (DocumentDB), and graph (Neptune) databases.
  - **Analytics Services:** Including services for data analysis at scale (Athena), big data processing (EMR), data integration (Glue), data warehousing (Redshift), real-time streaming (Kinesis), and business intelligence (QuickSight). Amazon Athena is highlighted for its ability to "*Analyze petabyte scale data where it lives (S3, for example)*".
  - **ML and AI Services:** Featuring a fully-managed ML platform (SageMaker) and pre-trained AI services for NLP (Comprehend), image/video analysis (Rekognition), text extraction (Textract), and machine translation (Translate).
- **Important Services for Data Analytics/Engineering:** EC2, Lambda, S3, RDS, DynamoDB, Glue, Athena, EMR, and Redshift are specifically highlighted as important for these domains.
- **AWS Free Tier:** A program offering hands-on experience with a subset of services for 12 months with certain limitations. Examples include free hours for EC2 and free storage and requests for S3.

Source 2: "12 - EC2 & Lambda.pdf"

Main Themes:

-

EC2 (Elastic Cloud Compute): Defined as "*Scalable Virtual Computing in the Cloud*," EC2 provides virtual servers (instances) with various types, a pay-as-you-go pricing model, and support for multiple operating systems.

- **Features of EC2:**
    - **Elasticity:** Ability to easily scale compute capacity up or down programmatically.
    - **AMIs (Amazon Machine Images):** Users can utilize standard AMIs or create and use their own pre-configured images.
    - **Integration:** Seamless integration with other AWS services like S3 and RDS.
- **EC2 Lifecycle:** Instances go through stages: Launch, Start/Stop (temporary suspension), Terminate (permanent deletion), and Reboot (restart without data loss on the root volume).
- **Data Storage Options for EC2:**
    - **Instance Store:** Temporary, high-speed storage tied to the instance's lifecycle.
    - **EFS (Elastic File System):** Shared file storage.
    - **EBS (Elastic Block Storage):** Persistent block-level storage.
    - **S3:** Used for large datasets and EC2 backups.
- **Common EC2 Use Cases:** Web hosting, data processing ("*It's a VM… you can do anything to data possible with a programming language.*"), machine learning (training models using GPU instances), and disaster recovery (backing up workloads and infrastructure).
- **Practical Example of Setting Up an EC2 Instance:** The excerpt outlines steps for launching an EC2 instance, connecting via SSH, updating packages, installing MiniConda and Streamlit, creating a basic Streamlit web application, and the need to open up the Streamlit port in the security group.
- **AWS Lambda:** Introduced as providing "*serverless computing*," Lambda allows running code in response to events without managing servers. Users "*only pay for execution time, not for idle compute time*".
- **Lambda Features:**
    - **Event-driven execution:** Triggered by various AWS events.
    - **Runtime support:** Supports multiple programming languages (Python, Java, Node.js, etc.).

  - **Integration:** Highly integrated with other AWS services.
  - **Scalability:** Automatically and rapidly adjusts to demand.
- **How Lambda Works:** Users upload code, configure event sources, and the Lambda function executes when an event occurs.
- **Basic Steps for Creating and Testing a Lambda Function:** The excerpt briefly mentions the process of creating a function, editing and deploying code, and testing it.

Key Takeaways:

- AWS is a comprehensive cloud platform with a vast array of services catering to diverse needs.
- Understanding the Shared Responsibility Model is critical for maintaining security in the AWS environment.
- EC2 provides flexible and scalable virtual compute instances for various workloads, requiring management of the underlying OS and infrastructure.
- Lambda offers a serverless computing model, abstracting away infrastructure management and providing cost-efficiency by charging only for execution time.
- Both EC2 and Lambda are fundamental services for data analytics and engineering tasks within the AWS ecosystem.
- The AWS Free Tier allows new users to explore and gain hands-on experience with many of the platform's core services within defined limits.