AWS Storage & Compute

Cloud Computing

Brenden west

Contents

Learning Outcomes

- Overview of AWS S3 file storage
- S3 operations & access control
- How to interact with S3 via AWS CLI or SDKs
- Overview of AWS cloud compute services
- Cloud compute with AWS EC2 & Elastic Beanstalk

Reading

- AWS Cloud Foundations - module 6, 7 (section 2)
- AWS Cloud Developing - module 3

Terminology

- **Bucket -**
- **Object** -

What is Amazon S3

- Object storage service offering scalability, availability, security, & performance
- Large-scale key-value data store, where values are file objects
- Objects can be any static content - e.g. documents, images, videos, audio, etc.
- S3 can generate event notifications when an object is created, deleted, restored, or replicated
- Can use storage policies to archive rarely used objects

AWS S3 Components

- **Bucket** - container for objects that organize content & identify the responsible account.
    - Each bucket lives in a specific region.
    - Can maintain object versions (off by default)
- **Object key** - unique identifier for an object
- **Object** - any type of file.
    - Can use name prefix to organize in pseudo folder structure (e.g. 2022/file1.txt)
    - Can be referred to by URL,
    - Consists of data & metadata (a set of name/value pairs that describe the object).
        - Metadata includes system-defined - e.g. creation date, size, version
        - Metadata can include user-defined values
    - Can be locked to prevent data changes

Creating S3 Buckets

- Buckets are created via Management Console, CLI, or AWS SDK
- Bucket is created in a specific region & region can’t be changed later
- Name must be globally unique
- Name must be 3-63 characters, lowercase, alphanumeric and hyphens only

Working with S3 Objects

- **PUT** - method to upload or copy an object to a bucket
  - Requires write permission
  - Must use multi-part upload for objects > 5 GB & should use for objects > 100 MB
- **GET** - method to retrieve objects from S3
  - Requires read access to the bucket
  - Can retrieve a complete object or a range of bytes (partial object)
  - Returns 404 for invalid or deleted objects
- **SELECT** - query S3 using SQL & return data for matching objects
  - Requires s3,GetObject permissions
  - Command includes a SQL expression and response format (e.g. JSON, CSV)
  - Can return all or specific data columns
- **DELETE** - method to permanently delete one or more objects
  - Need to specify version ID in case of versioned objects

Securing S3 Data

- **Encryption** - data can by encrypted in transit & at rest
    - At rest encryption can be managed by client or by AWS S3 or KMS
- **Identity-based policies** - grant permissions to users, groups, or roles in same AWS account
- **Access control lists (ACLs)** - resource-based policies that manage access to buckets and objects
- **Bucket policies** - resource-based policy to supplement or replace ACLs.
    - Can grant permissions to other AWS accounts or IAM users
    - Object permissions apply only to objects the bucket owner creates
- **Pre-signed URLs** - provide PUT or GET access using permissions of user who created the URL
- **Cross-origin resource sharing (CORS)** - specific rules to identify origins that can access a bucket & allowed operations

AWS Cloud Compute

AWS offers a wide range of cloud services for running ‘compute’ applications.

The most common for general-purpose computing solutions are:

- **AWS Elastic Cloud Compute (EC2)** - runs virtual machines
- **AWS Elastic Beanstalk** - a fully-managed service for running applications
- **AWS Elastic Container Service (ECS)** - runs containerized applications
- **AWS Fargate** - fully managed service to run containers
- **AWS Lambda** - A serverless compute service to run applications

AWS also provides a number of services to support these compute solutions

AWS Compute Services

Services	Key Concepts	Characteristics	Ease of Use
• Amazon EC2	• Infrastructure as a service (IaaS) • Instance-based • **Virtual machines**	• Provision virtual machines that you can manage as you choose	A familiar concept to many IT professionals.
• AWS Lambda	• **Serverless** computing • Function-based • Low-cost	• Write and deploy code that runs on a schedule or that can be triggered by events • Use when possible (architect for the cloud)	A relatively new concept for many IT staff members, but easy to use after you learn how.
• Amazon ECS • Amazon EKS • AWS Fargate • Amazon ECR	• **Container-based** computing • Instance-based	• Spin up and run jobs more quickly	AWS Fargate reduces administrative overhead, but you can use options that give you more control.
• AWS Elastic Beanstalk	• Platform as a service (PaaS) • For **web applications**	• Focus on your code (building your application) • Can easily tie into other services—databases, Domain Name System (DNS), etc.	Fast and easy to get started.

AWS EC2

- Provides virtual machines (IaaS) based on an Amazon Machine Image (AMI)
- Allows developers flexibility & full administrative control
- Developers can OS and machine capabilities
- Developers can run any number of instances of the same AMI
- Developers can increase or decrease size of existing servers
- EC2 instances can be defined & launched via the AWS console, CLI, or SDKs

Amazon Machine Images - AMI

- An AMI is a virtual machine template that specifies OS and any software installed on that OS (e.g. Linux, Python, etc.)
- Amazon provides a number of pre-defined AMIs for popular configurations
- Developers can choose from 3rd-party or community-provided AMIs or upload their own AMIs
- Developers can configure an EC2 instance and save it as an AMI
- An AMI is required to launch an EC2 instance

AWS EC2 Instance Types

- AWS ECS supports a range or pre-defined instance types that vary according to:
  - Memory (RAM)
  - Processing power (CPU)
  - Disk space and disk type (Storage)
  - Network performance
- Instance types are categorized according to intended use:
  - General purpose
  - Compute optimized
  - Memory optimized
  - Storage optimized
  - Accelerated computing

AWS EC2 Instance Types, cont.

- EC2 instances are defined by family (e.g. T), generation and size (e.g. nano, micro, small, medium, large)
- Size is a pre-defined combination of CPU, memory, and network bandwidth

EC2 Instance Configuration

Before launching an EC2 instance, you must specify

- Network location where the instance will be deployed
- Whether the instance will have a public IP address
- Security groups for incoming & outgoing network traffic

Optional configurations include:

- IAM role for the instance to communicate with other AWS services
- User data - scripts to run when the instance first starts
- Additional storage volumes (size, type, & whether storage is retained)
- A key pair to allow secure connection to the running instance

AWS EC2 Storage options

- **Amazon Elastic Block Store** - durable block-level storage
- **Amazon EC2 Instance Store** - Ephemeral storage attached to host computer where instance is running
- **Amazon Elastic File System** - a simple, scalable, fully managed elastic
- Network File System (NFS) file system
- **Amazon Simple Storage Service (Amazon S3)** - an object storage service that offers scalability,data availability, security, and performance.

-

Amazon EC2 Pricing Models

Developers should understand Amazon’s different EC2 pricing models:

- **On-Demand Instances** - eligible for the AWS Free Tier. Have the lowest upfront cost and the most flexibility. There are no upfront commitments or long-term contracts. A good choice for applications with short-term, spiky, or unpredictable workloads.
- **Dedicated Instances** - physical services with instance capacity dedicated to a single customer & physically isolated from other AWS accounts
- **Dedicated Hosts** - physical servers with instance capacity allowing use of existing software licences

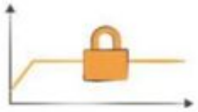
Amazon EC2 Pricing Models, cont.

- **Reserved Instances** - reserved for specific term (e.g. 1 yr, 3 yr) for a fixed, discounted price. Good for consistent long-term needs.
- **Scheduled Reserved Instances** - similar to reserved instances, but for a specific, recurring duration.
- **Spot Instances** - customer can bid on unused EC2 instances. Spot instance runs whenever bid exceeds current market price. Can be terminated on short notice.

Cost Optimization: Principles

- **Right size** - choose right balance of instance types & allow for servers to be sized down or turned off. Use monitoring to identify idle capacity.
- **Increase elasticity** - design deployments to scale for peak load and minimize idle server capacity. Use automatic scaling to match needs.
- **Optimal pricing model** - match usage to appropriate pricing option
- **Optimize storage choices** - choose least expensive storage options that match usage requirements
- **Measure, monitor, & improve**

Amazon EC2 pricing models: Benefits



On-Demand Instances	Spot Instances	Reserved Instances	Dedicated Hosts
• Low cost and flexibility	• Large scale, dynamic workload	• Predictability ensures compute capacity is available when needed	• Save money on licensing costs • Help meet compliance and regulatory requirements