AMAZON WEB SERVICES:
THE DEFINITIVE GUIDE FOR BEGINNERS AND ADVANCED USERS

Parul Dubey
Arvind Kumar Tiwari
Rohit Raja

Bentham Books

Amazon Web Services: the Definitive Guide for Beginners and Advanced Users

Authored by

Parul Dubey
Department of Artificial Intelligence
G H Raisoni College of Engineering
Nagpur, India

Arvind Kumar Tiwari
Department of CS & IT
Dr. C V Raman University
Bilaspur, India

&

Rohit Raja
Department of Information Technology
Guru Ghasidas Viswavidyalaya (Central University)
Bilaspur, India

Amazon Web Services: the Definitive Guide for Beginners and Advanced Users

Authors: Parul Dubey, Arvind Kumar Tiwari and Rohit Raja

ISBN (Online): 978-981-5165-82-1

ISBN (Print): 978-981-5165-83-8

ISBN (Paperback): 978-981-5165-84-5

First published in 2023.

need for a court order if at any point you breach any terms of this License Agreement. In no event will any delay or failure by Bentham Science Publishers in enforcing your compliance with this License Agreement constitute a waiver of any of its rights.

3. You acknowledge that you have read this License Agreement, and agree to be bound by its terms and conditions. To the extent that any other terms and conditions presented on any website of Bentham Science Publishers conflict with, or are inconsistent with, the terms and conditions set out in this License Agreement, you acknowledge that the terms and conditions set out in this License Agreement shall prevail.

Bentham Science Publishers Pte. Ltd.
80 Robinson Road #02-00
Singapore 068898
Singapore
Email: subscriptions@benthamscience.net

CONTENTS

PREFACE

Welcome to Amazon Web Services: the Definitive Guide for Beginners and Advanced Users! We are excited to share our knowledge and expertise with you through this comprehensive guide.

As cloud computing continues to transform the way we do business, Amazon Web Services (AWS) has emerged as a leader in the field, providing a vast array of services and tools that can help organizations of all sizes to innovate, streamline operations, and accelerate growth. However, navigating the complex world of AWS can be daunting, especially for those who are new to the platform.

In this book, we have endeavoured to provide a complete guide to AWS that is accessible to both beginners and advanced users. Whether you are just starting out with AWS, or you are looking to deepen your understanding of advanced topics such as server less computing, machine learning, and containerization, this book has something for you.

Throughout the book, we provide clear explanations and practical examples that illustrate how to use AWS services effectively, as well as tips and best practices for optimizing performance, reducing costs, and ensuring security. We also cover the latest updates and news features in AWS so that you can stay up-to-date with the rapidly evolving landscape of cloud computing.

Whether you are an entrepreneur, a software developer, a system administrator, or a data analyst, this book will help you to master AWS and unlock the full potential of cloud computing. We hope that you find this guide to be an invaluable resource, and we look forward to hearing about the ways in which AWS is transforming your business.

Thank you for choosing this book, and happy reading!

Parul Dubey
Department of Artificial Intelligence
G H Raisoni College of Engineering
Nagpur, India

Arvind Kumar Tiwari
Department of CS & IT
Dr. C V Raman University
Bilaspur
India

&

Rohit Raja
Department of Information Technology
Guru Ghasidas Viswavidyalaya (Central University)
Bilaspur
India

CHAPTER 1

Introduction To Cloud Computing and AWS

Abstract: This chapter offers an in-depth exploration of cloud computing, encompassing various deployment models and services. It sheds light on the pressing concern of personal data privacy, as cloud technology enables data to traverse geographical boundaries, disregarding local regulations. The evolution of AWS since 2000 takes the center stage, emphasizing key elements such as regions, availability zones, and data centers. By providing comprehensive insights, the chapter equips users with the necessary knowledge to effectively navigate the intricacies of AWS. It also elucidates the AWS shared responsibility model, ensuring users understand their own responsibilities and those of AWS. Furthermore, the chapter focuses on six core competencies, enabling customers to fully harness the potential of AWS's offerings. This multifaceted chapter serves as a comprehensive guide, empowering readers to grasp the complexities and opportunities of cloud computing with AWS.

Keywords: AWS evolution, Core competencies, Cloud computing, Deployment models, Data privacy, Shared responsibility model.

INTRODUCTION

The phrase “cloud computing” is used to describe several different approaches, but they all have the common goal of providing hosted services through the internet. The infrastructure, platform, and software as a service are the three main parts of any given cloud computing service (SaaS).

Public clouds and private clouds are two different kinds of cloud computing. Customers can buy services from a public cloud provider over the internet [1, 2, 3]. To provide hosted services to a limited audience, a network or data center might function as a “private cloud.” Access to the network and the kind of actions that may be performed with its resources are strictly controlled for these users. The purpose of cloud computing is to increase accessibility and scalability of IT resources and services by placing them in a central location accessible through the internet or another network.

To initiate a cloud computing paradigm, one needs both the proper software and hardware components. The term “cloud computing” is commonly used interchangeably with “utility computing” and “on-demand computing.”

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

Hundreds of thousands of companies in over 190 countries have made AWS their go-to cloud infrastructure platform. It has a solid reputation for dependability, scalability, and affordability.

CLOUD ARCHITECTURE

Cloud computing is used by companies of all sizes to store data and make it accessible to workers from any place in the company as long as they have an internet connection [4, 5, 6]. The architecture of cloud computing is a combination of ideas known as service-oriented and event-driven architectures [7, 8]. Fig. (**1**) explains the architecture of the cloud. The following are the two primary components that make up the architectural framework of the cloud:

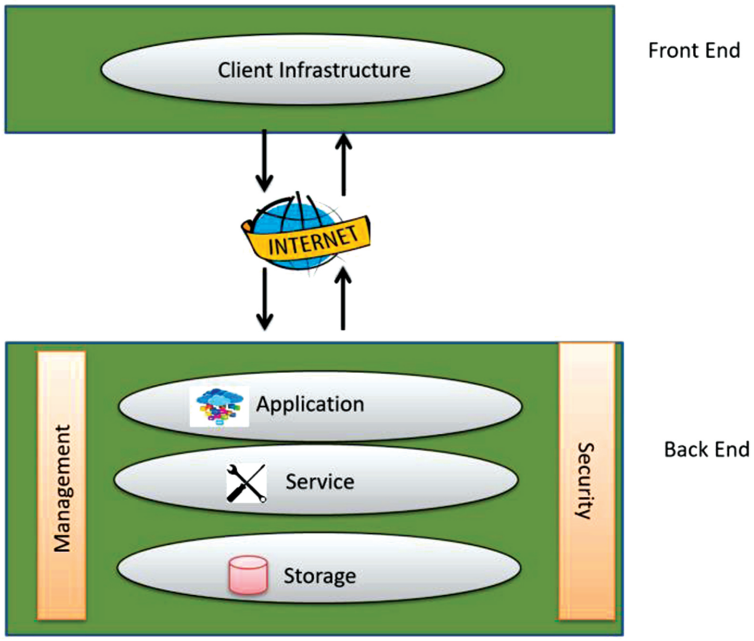


Fig. (1). Architecture for Cloud computation.

- **Front end:** The client works with the front end. It has the client-side APIs and applications needed to connect to cloud-based systems. Thin and fat clients, tablets, and mobile devices make up the front end, together with web servers (such as Chrome, Firefox, Internet Explorer, *etc.*).
- **Back end:** Service providers often make use of the back end. It controls every component needed to provide cloud computing services. Storage for vast quantities of data, firewalls, virtual machines, deployment methods, servers, traffic management systems, *etc.* are all part of the picture.

TYPES OF CLOUD COMPUTING SERVICES

There are three main categories into which services provided by the cloud may be categorized [7]. An overview of various categories of cloud services is shown in Fig. (**2**).

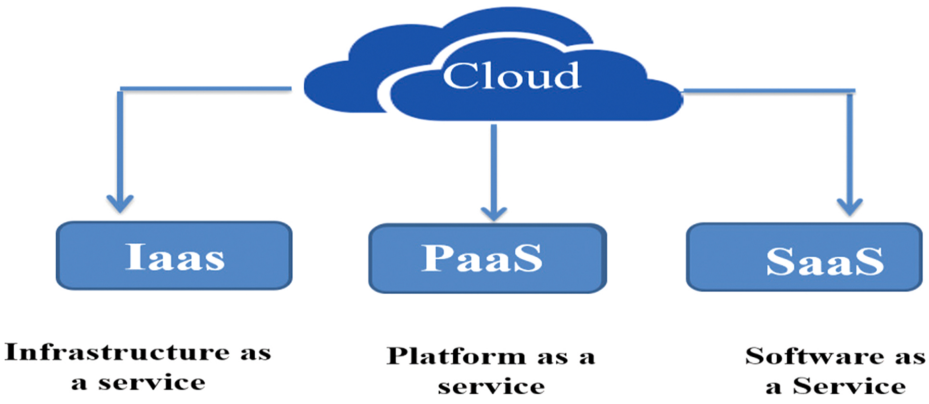


Fig. (2). Types of cloud services.

Infrastructure as a Service (IaaS)

When we use infrastructure as a service, a cloud provider handles the nitty-gritty details on their behalf. These details include the management of servers, networks, virtualization, and data storage. Access to and management of the underlying infrastructure may be provided to the user in the form of either an application programming interface (API) or a control panel. User responsibility is restricted to the operating system, applications, and middleware, while provider supervision encompasses the hardware, networking, hard drives, data storage, and servers in addition to the management of any interruptions, repairs, or physical issues that may occur. This is the standard architecture for data storage services that are hosted in the cloud.

Platform as a Service (PaaS)

A model of cloud computing known as platform as a service (PaaS) is one in which the underlying infrastructure and application software are supplied and managed by a third-party provider. The user, on the other hand, is responsible for managing the applications that are deployed on top of that infrastructure as well as the data on which they rely. PaaS is primarily used by developers and programmers, and it gives these individuals access to a shared cloud platform for application development and management (an essential component of devops),

CHAPTER 2

Identity and Access Management in AWS

Abstract: This chapter provides a comprehensive exploration of Identity and Access Management (IAM) in AWS. It delves into the detailed processes of creating IAM users, roles, and setting policies, emphasizing their significance in safeguarding the AWS account and its users. IAM proves invaluable in managing multiple users within a single account. The chapter also highlights the crucial role of Multi-Factor Authentication (MFA) in enhancing account security, particularly when collaborating with AWS Cloud Service Providers (CSP). Overall, this chapter serves as a fundamental guide for understanding and implementing IAM, establishing a strong foundation for working effectively within the AWS ecosystem.

Keywords: Account security, Identity and access management (IAM), IAM users, IAM roles, IAM policies, Multi-Factor authentication (MFA).

INTRODUCTION

Through the use of AWS Identity and Access Management (IAM), it is able to control access to AWS services in a safe manner [55]. Customers of Amazon Web Services (AWS) may regulate who is authenticated and granted permission to access resources by using IAM.

Customers are granted access to all of AWS's services and resources *via* the use of a single sign-in identity when they first sign up for an account with Amazon Web Services (AWS). Accessing this account is as simple as logging in with the email address and password that customers used when they first set up their AWS account. It is highly recommended that we refrain from doing administrative responsibilities when logged in as the root user. Instead, it is advised that users make use of the root user by themselves in order to generate the very first IAM user. After then, the information required to log in as root is utilized in order to administer a few of the system's accounts and services.

IAM CAPABILITIES

The following capabilities are made available to users by IAM:

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

- **Access to the shared AWS account:** AWS users have the option of granting access to their account without disclosing their credentials or access key to other parties.
- **Permissions at various levels:** One may offer different individuals differing permissions for various resources. One may, for example, grant specific users complete access to AWS services. Users are only permitted to provide other users read-only access to certain S3 buckets, permission to manage specific EC2 instances or visibility into their billing information; nothing else is allowed.
- **Preventing Amazon EC2 from accessing AWS resources:** IAM capabilities may be used to deliver credentials to applications that are executing on EC2 instances in a safe manner. The application will now be able to access more AWS resources owing to these credentials. One situation of this would be S3 buckets.
- **Multi-factor authentication (MFA):** Users have the option of enabling two-factor authentication for both the account as a whole and individual users within it. This provides an additional layer of safety. In order to access the account when it is protected by MFA, users will need to provide not just a password or access key, but also a code generated by a device that is suitably equipped.
- **Identity Federation:** Access to the IAM system may be granted based on the trust placed in an existing form of authentication, such as a Facebook or Google account. Using the same credentials for both on-premises and cloud-based operations simplifies password management.
- **Free of cost:** IAM security is included at no extra cost. Adding new users, groups, or policies does not incur any extra fees.
- **In line with PCI DSS:** An information security standard for businesses that deals with credit cards issued by major credit card companies is the Payment Card Industry Data Security Standard (PCI DSS). IAM is in line with this requirement.
- **Authentication policy:** Using the IAM password policy, one may change or reset a user's password from any location. A user may be restricted by a set of constraints, such as how many times he or she can try to enter a password until being locked out.

WORKING OF IAM

The IAM process may be broken down into these six steps:

1. **Principal:** A person or organization that has access to an Amazon Web Services (AWS) resource is referred to as a principal. A user, a job, or an application might all qualify as a principle in this context.

2. **Authentication:** Verifying the identity of the user who is seeking to log in to an Amazon Web Services account is required in order to gain access to any of those services. The provision of the required credentials or keys by the principal is essential to the success of the authentication process [55].

3. **Request:** A request is sent to AWS by a user, who identifies both the activity that should be performed and the resource that should do it.

4. **Authorization:** By default, authorization is not granted for any of the resources. Before a request can be granted by IAM, each component of the request must first get approval from a matching policy. When the request has been authenticated and authorized, AWS will take it into consideration and accept it.

5. **Actions:** Through the use of actions, a resource may be accessed, generated, changed, or removed.

6. **Resources:** A collection of tasks that are unique to the Amazon Web Services (AWS) account may be completed with the assistance of Amazon Web Services (AWS) resources.

CONSTITUENTS OF THE IAM

IAM consists of a number of other key parts in addition to these. First, we focus on the individual user, who is a component of a bigger group [56]. Policy engines are the components that are accountable for deciding whether or not a connection is allowed. A role is a credential that is only valid for a certain amount of time and may be issued to an instance at any point in time [57].

- **Users:** An individual who has been granted a certain set of credentials and the permissions that go along with them is referred to as an IAM user. Depending on the point of view, a user might be a real person or a software. By assigning each member of the team their very own distinct IAM user name, one may utilize IAM to govern who is permitted to use the services provided by AWS. There is only one AWS account associated with each individual IAM user [58]. A newly created user in AWS is not granted permission to do anything when the account is first created. Having a one-to-one user specification model is advantageous since it allows for the individual assignment of permissions to each user. Fig. (**1**) shows the IAM dashboard once we login into the AWS account and this page mentions all necessary details like the number of groups, and users and it doesn't require any region selection as it's a global service. Details of how to create an IAM user are given in the next section.

CHAPTER 3

Amazon EC2 Instance

Abstract: This chapter offers comprehensive coverage of EC2 (Elastic Compute Cloud) in AWS, encompassing instance types, pricing, and other essential aspects. It provides detailed step-by-step instructions for setting up an EC2 instance and initiating work on the AWS platform. By following these procedures, readers gain the necessary knowledge and practical guidance to effectively utilize EC2 and leverage other capabilities offered by AWS. This chapter serves as a valuable resource for individuals seeking to navigate the intricacies of EC2 and fully harness the potential of AWS services.

Keywords: Amazon machine images, Elastic compute cloud, EC2 instance connect, Instance types, Pricing.

INTRODUCTION

An Amazon Elastic Compute Cloud (EC2) instance is a virtual machine for running applications on the Amazon Web Services (AWS) architecture. EC2 is a service that enables business customers to leverage the cloud computing environment to run applications. It may serve as a practically unlimited group of virtual machines (VMs).

Amazon provides a large choice of instance types with customized Processor, memory, space, and connectivity options. Each sort is produced in various sizes to satisfy diverse task demands.

Amazon Machine Images are utilized to deliver instances (AMI). The machine images are like templates. The OS and other software placed on them set the setting for the user's experience. Users may choose an AMI given by AWS, the user community or through the AWS Marketplace. Users may also create their own AMIs and share them to other users.

TYPES OF EC2 INSTANCES

Instance types are organized into groups depending on intended application characteristics [60, 61]. The naming of the instance in AWS is given in Fig. (**1**). The following are examples of these categories:

Fig. (1). Naming of an Instance in AWS.

- **General purpose:** A general-purpose instance is a virtual machine (VM) that can function under a wide range of conditions. General-purpose instances are designed to have a large number of CPU cores, on-demand storage and RAM. Web server hosting and application testing and development are two typical applications of general-purpose instances.
- **Compute optimized:** On the AWS cloud, compute optimized instances are used to execute big data applications that need a lot of memory and processing capacity. These instances were built from the ground up to handle the most demanding workloads, including those that process large amounts of data and need high rates of input and output operations per second (IOPS). Scientific and financial modeling and simulation, machine learning, corporate data warehousing, and business intelligence are all examples of the sorts of applications available.
- **Graphics processing unit (GPU):** Applications that rely heavily on graphics may now be performed much more quickly than before on these instances, as opposed to using the more basic EC2 instances. Both gaming and design software use graphics processing units (GPUs). GPUs are often used by Linux distributions for a variety of tasks, including graphical user interface rendering, data compression, and database query optimization.
- **Memory-optimized:** In order to offer lightning-fast data access and exceptional performance, memory-optimized instances rely on solid-state drives. These instances shine when it comes to tasks like in-memory caching, real-time big data analytics, and running open-source databases.
- **Storage optimized:** I/O-intensive applications, such as NoSQL stores that save and extract data in real-time, benefit greatly from storage optimized instances. They're also ideal for memory-intensive applications including processing data, data warehousing, analytics workloads, and log processing.
- **Micro:** Application workloads that need low throughput should use a micro instance. The micro instance type may function as a tiny host computer, as a platform for testing processes or as a web server that does not need large transaction rates.

PRICING STRATEGY OF EC2 INSTANCE

Amazon Web Services gives us a number of different purchase choices for EC2 instances so that one may service their workload in the most efficient and cost-effective way possible [62].

On-demand Instance

- Getting an EC2 instance up and running on Amazon Web Services may be accomplished in a number of different ways, but this is one of the most flexible methods. Each launch of computational power will result in a charge to the account in seconds [63].
- Individuals are not required to do anything, and it is totally okay for users to terminate the instance at any time after the first 5 minutes have passed.
- When a process is first launched when it is paused, when it is hibernated, when it is restarted, and when it is finally terminated is totally up to the business. When the instance is up and running, customers will only be charged for the amount of time it is really functioning as intended. A calculation of the cost per second may be seen in this section [64].
- There is no guarantee that Amazon Web Services will actually launch the instance. Ever though AWS will always make an effort to create the instance, practically its very rare in a circumstance where we request an on-demand instance and it won't launch. Despite this, AWS may still encounter peak demand if the instance we want to use is not launched at the time we want it to be started.
- This is the greatest approach for dealing with unpredictable short-term or irregular workloads that can't be halted.
- Users are free to use it in the process of developing and testing applications.

Reserved Instances

- If we use Reserved Instances instead of on-demand EC2 instances, one might save as much as 72 percent of the total cost. Before we get started, all one needs to do is decide on a certain instance configuration, instance type, and duration (either one year or three years) [64].
- Ideally suited for continuous operation, such as a server.
- Make use of this option in order to establish a time commitment and configuration specification for the reserved EC2 instances.
- Users have the option of reserving a property for either one or three years in advance. There is no range from one to three years; it is either one or three years. There is no other option available.

CHAPTER 4

Storage Options with Amazon EC2 Instance

Abstract: This chapter provides an overview of the storage options available with EC2 instances, namely EC2 Instance storage, EBS, EFS, and Amazon S3. It focuses on illustrating the creation of EBS volumes and attaching them to EC2 instances. The chapter highlights the unique features and uses cases of each storage option, empowering readers to make informed decisions based on their needs. By following the step-by-step instructions, readers gain practical guidance on configuring and utilizing EBS storage effectively within the EC2 environment. Overall, this chapter serves as a valuable guide for optimizing storage configurations and maximizing the benefits of EC2 instances in AWS.

Keywords: Amazon S3 (Simple Storage Service), EC2 instance storage, EBS (Elastic Block Store), EFS (Elastic File System), Storage options.

INTRODUCTION

Instances are capable of making use of the adaptable, user-friendly, and cost-effective data storage options that Amazon EC2 provides. There is a wide range of efficiency and durability among the available choices. Depending on the particular requirements at hand, an individual may use one or more of these various storage systems alone or in combination.

We are able to link permanent, block-level storage volumes to running instances with the help of Amazon Elastic Block Store, also known as EBS. Amazon Elastic Block Storage (EBS) is a potential candidate for the role of primary storage medium for data that requires frequent and fine-grained updates. Amazon Elastic Block Storage (EBS) is the recommended storage solution to use in many scenarios, such as when a database is being operated on an instance.

Users have the option of using Amazon Elastic File System in order to rapidly build, conveniently extend, and optimally optimize their file storage on Amazon Web Services. With just a few mouse clicks in the AWS Management Console, a user is able to make file systems accessible to Amazon EC2 instances, Amazon container services (ECS, EKS, and AWS Fargate), and AWS Lambda functions by using a file system interface. These file systems can be made available *via* the

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

AWS File System Interface. Complete access semantics to the file system is implemented, including strong consistency and file locking.

STORAGE

Amazon Elastic Compute Cloud (EC2) provides users with a broad range of options that are not only inexpensive but also straightforward and flexible when it comes to storing data associated with the instances. There is a range of possibilities, each with its own level of effectiveness and durability. One may utilize any one of these ways of storage on its own, or they can mix many strategies to better suit their needs [66].

After going through this section, users should have a solid understanding of how to adjust the data storage options in Amazon EC2 to suit their requirements [66]. The following Fig. (**1**) illustrates how these different types of storage relate to the specific instance. The following are some of the different ways of storing that are available:

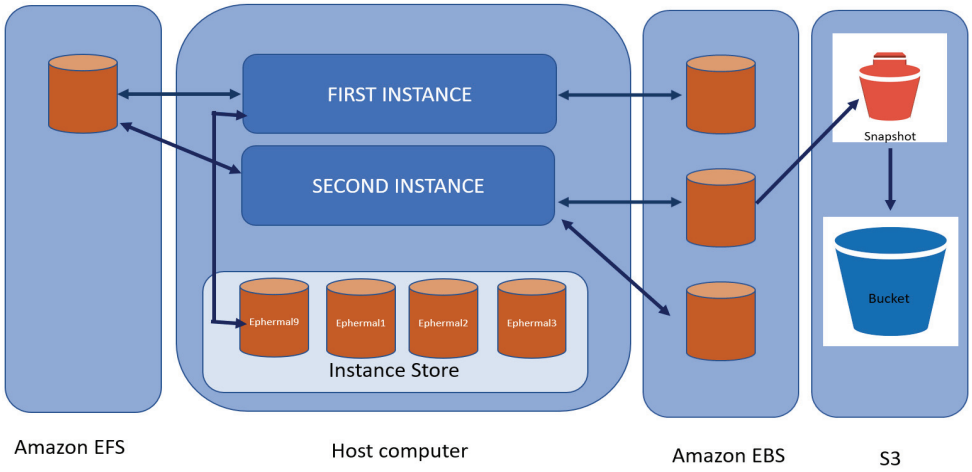


Fig. (1). Different storage types.

- Amazon EC2 instance store
- Amazon Elastic Block Store(EBS)
- Amazon Elastic File Storage (EFS)
- Amazon Simple Storage Service (S3)

First three will be discussed in this chapter. Amazon S3 will be discussed in detail in chapter 6.

AWS EC2 INSTANCE STORAGE

EC2 Instance Storage is the name of the service that Amazon Web Services (AWS) provides for its short-term block storage solution [67]. Instance storage in Amazon Elastic Compute Cloud (EC2) is not a storage service in and of itself; rather, it is an essential part of the EC2 offering. This sort of storage is physically situated on the same host that supplies the EC2 instance, which makes it a great choice for keeping transitory information connected to EC2 instances. Additionally, this kind of storage is quite affordable.

The instance storage volumes can only be set when the instance is first started. It is not possible to disconnect a volume from one instance and then attach it to another instance because instance store volumes are tied to their instances.

The contents of an instance store are only accessible for the length of time that corresponds to the instance to which they are attached [67, 68]. Data stored in instances survives a reboot, regardless of whether or not the reset was scheduled. Any of the following causes data loss in the instance store:

- Disk drive failure occurs at the system level
- stopping the instance
- A hibernating instance
- terminating the instance

As a result, we should not store any data that is important or long-term in the instance storage. Consider storing company's data using Amazon S3, Amazon EBS, or Amazon EFS for improved long-term access and reliability.

When an instance is stopped, hibernated, or terminated, each and every piece of data that was previously stored in the instance store is deleted. As a result, the data cannot be accessed by any other instances using the instance's store.

When a new instance is started from an AMI, the information that is saved on the volumes of the instance store of the previous instance is not transferred to the volumes of the new instance, and *vice versa*. After users apply this change, the instance store will be severed from its connection to the newly created instance type. EC2 instance storage features include the following:

- Short-term storage- Data that is only needed temporarily may be saved in the EC2 instance storage.
- Cost- The EC2 instance itself is the one responsible for paying for these storage volumes. Despite the fact that the storage capacity of each instance is different,

CHAPTER 5

Load Balancing and Auto Scaling in AWS

Abstract: This chapter provides a thorough introduction to Autoscaling and Elastic Load Balancing (ELB) in the AWS cloud. It highlights the benefits of these services, which simplify workflows and automate tasks. The chapter offers an in-depth overview of ELB and Autoscaling, covering their basic uses and providing hands-on experience. By understanding and utilizing these services, users can optimize resource allocation, handle traffic fluctuations, and enhance application performance. This chapter serves as a comprehensive guide for effectively implementing Autoscaling and Elastic Load Balancing in the AWS environment.

Keywords: Autoscaling, Application load balancer, Elastic load balancing (ELB), Network load balancer, Resource allocation.

INTRODUCTION

Using Elastic Load Balancing, the incoming traffic to any EC2 instances, containers, or IP addresses located in one or more Availability Zones is automatically spread over all of those zones. It monitors the status of the registered targets and only sends traffic in their direction if those targets are online and able to process it. The capacity of the load balancer may be dynamically increased or decreased by Elastic Load Balancing in response to changes in the amount of incoming traffic.

TYPES OF LOAD BALANCERS

Elastic Load Balancing is compatible with Application Load Balancers, Network Load Balancers and Gateway Load Balancers. Classic Load Balancers were the class of load balancers that is outdated [72, 73, 74, 75]. We have the option of selecting the sort of load balancer that is most suitable for our needs.

Application Load Balancers

We need a load balancer that can select where to send requests at the application layer (HTTP/HTTPS), open up specified ports on each container instance, and utilize path-based routing so that we can spread traffic throughout the cluster's

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

containers. Application load balancers often come equipped with a capability that allows for dynamic host port mapping. A dynamic selection of the host port will be made from the container instance's range of temporary ports if the task's container specification for a NGINX container port is port 80 but the host port is 0. When the job is launched, the Application Load Balancer obtains the instance ID and port of the NGINX container and registers them as a pair [76]. After this, the traffic is directed to the instance ID and port of the NGINX container. Because this mapping is so flexible, it enables one to execute several tasks from the same service all from inside the same container.

Network Load Balancer

A Network Load Balancer's operations take place at the transport layer (TCP/SSL), which is where routing decisions are determined. It can handle millions of inquiries in a single second without a hitch. The load balancer makes use of a flow hash routing method whenever a connection is formed in order to determine, in accordance with the rule that is in place by default, which member of the target group will be the beneficiary of the connection. It uses the port number that has been configured for the listener in order to make an attempt at establishing a TCP connection with the host that has been selected. Before the request is sent any further, the headers of the request are not changed in any way. Network load balancers come equipped with a function known as dynamic host port mapping [75, 77]. If we set an NGINX container port to 80 and a host port to 0, then the container instance will select a random port from the range of accessible ports when it is being created on the fly (such as 32768 to 61000 on the latest Amazon ECS-optimized AMI). The Network Load Balancer obtains the instance ID and port of the NGINX container the moment the task is started and then correlates those two pieces of information with a load balancing group. Because of the existence of this dynamic mapping, it is now possible to execute many tasks from the same service inside a single container.

Gateway Load Balancers

Gateway load balancers make it possible to install, scale, and administer a wide variety of virtual appliances, including firewalls, intrusion detection and prevention (IDS/IPS), and data intrusion prevention (DPI) systems. It simultaneously scales the virtual appliances to meet demand while distributing traffic, and it integrates a transparent network gateway (a single point of entrance and exit for all network traffic) [78]. At Network Layer 3 of the OSI model is where we can find a Gateway Load Balancer. It checks any open ports for incoming IP packets and then sends them on to the specified group after monitoring those ports. Flows are maintained inside a single device by using a 5-

tuple (for TCP/UDP flows) or a 3-tuple (for flows that are not TCP/UDP), respectively. The Gateway Load Balancer makes use of the GENEVE protocol, which is located on TCP port 6081, in order to connect with other programs [79]. This makes it possible for the Gateway Load Balancer to connect with the instances of virtual appliances that have been registered. It is only able to process data packets that have a maximum of 8500 bytes in size. Through the use of Gateway Load Balancer endpoints, it is possible to exchange traffic securely across VPC boundaries. The Gateway Load Balancer is one example of a VPC endpoint, and its primary function is to provide encrypted communication between virtual appliances located in a service provider VPC and application servers located in a service consumer VPC. In addition to the other virtual appliances, a Gateway Load Balancer is installed in the private cloud that is being used. Users are able to link the virtual appliances with a certain user or group while using the Gateway Load Balancer.

HANDS-ON ELB

In this section, we shall work with Elastic Load Balancer. Here we will be hosting a web page with the help of two EC2 instances and accessing the webpage with the help of an Application load balancer (as it makes use of http request). The steps for the same are discussed below:

Launching of EC2 instances with hosting a webpage on it. Below mentioned steps are to be followed for this:

1. For this we need to launch two instances, following the steps discussed earlier in the section with EC2 instance launching. This will begin as shown in Fig. (**1**).

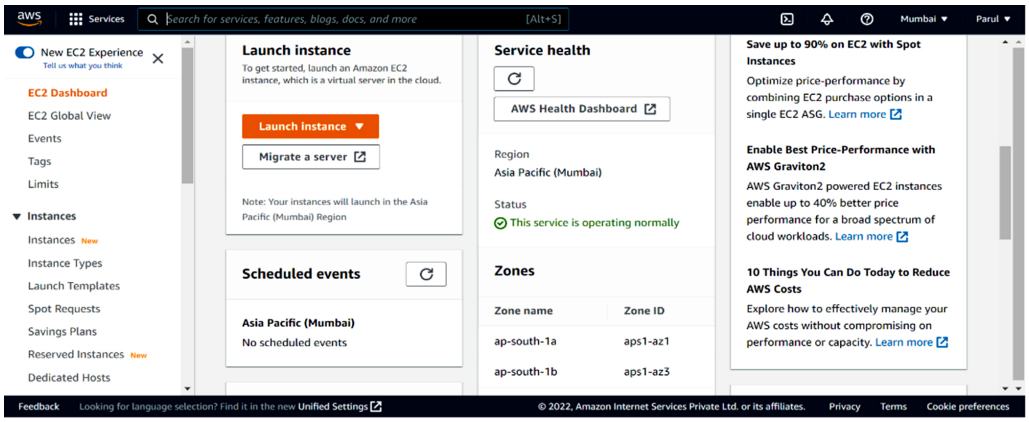


Fig. (1). Launching of Instance.

CHAPTER 6

Amazon Simple Storage Service

Abstract: This chapter provides a comprehensive overview of Amazon S3, a crucial service offered by AWS. It emphasizes the practicality of working with AWS and highlights the importance of understanding and utilizing Amazon S3 in various projects. The chapter covers the basics of AWS S3, including fundamental concepts and functionalities. Additionally, it explores advanced topics such as versioning and different Amazon S3 storage classes, enabling readers to maximize the potential of S3 in future endeavors. By equipping readers with the necessary knowledge and skills, this chapter serves as a valuable resource for effectively utilizing Amazon S3 within the AWS ecosystem.

Keywords: Amazon S3, Object, Storage classes, S3 bucket, Versioning.

INTRODUCTION

The Amazon Simple Storage Service (S3) is a cloud-based data storage and retrieval system. It's a cheap and straightforward storage service with unmatched longevity, accessibility, performance, security, and scalability.

A user-friendly web service interface is provided by Amazon Simple Storage Service (S3), which enables users to store data and retrieve it at any time and from any location [66, 71, 81]. Using this service, it is straightforward to develop programmes that utilise cloud storage in a manner that is as close to native as possible. The scalability of Amazon S3 and the pay-as-you-go pricing mechanism ensure that the performance and uptime of the application will not be negatively impacted despite its growing size.

Amazon S3 was designed with flexibility in mind from the beginning of its development. We are able to develop from a simple FTP client a big web application such as Amazon's online bookstore, to preserve any amount of data, which reads the same piece of data a million times or simply in case of emergency catastrophe recovery, and a great deal more. Since Amazon S3 takes care of the storage of data, programmers are free to focus on developing new features rather than finding out where to place the data they've created.

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

S3 BUCKETS

Before we can begin uploading our data to Amazon S3, we will need to begin by creating an S3 bucket in one of the AWS Regions.

In Amazon S3, a folder that can be used to store items is referred to as a "bucket" [82]. The user account can have a maximum of one hundred buckets, each of which has an infinite capacity for the number of objects it can hold.

AWS resources such as buckets and objects can be handled by utilizing the Amazon S3 APIs in various ways. Using the Amazon S3 Application Programming Interface (API), a user is able to do actions such as creating a "bucket" and storing objects within it. An alternative is the Amazon S3 administrative console, which can be used. The Amazon S3 Application Programming Interfaces (APIs) are utilized in order to send queries to Amazon S3 *via* the console.

It is required that each bucket name within an Amazon S3 partition be unique across all AWS accounts and over all AWS Regions.

No other AWS account that is part of the same partition is permitted to use the name until the bucket has been deleted. Do not place confidence in the naming convention of a bucket to determine whether or not it is available or safe.

Users of Amazon S3 have the option to have buckets produced in the selected area while utilizing the service. For users that have specific requirements regarding latency, cost, or compliance, selecting any AWS Region that is physically close to them is a smart alternative. It's possible that folks who live in India will find it helpful to make buckets in Mumbai.

S3 OBJECTS

Amazon Simple Storage Service (S3) is an object store that allows us to store an unlimited number of objects, each of which is identified by a unique key and value. Each object stored in a user's bucket(s) can be up to 5 terabytes in size [83]. In general, an object has the following parts:

- Key- A thing's label is the name we give it. For this purpose, we resort to the object key.
- Version ID- A key and version ID together provide the unmistakable identity of an object within a bucket. When we store something in a bucket on Amazon S3, the service assigns it a unique identifier called a "version ID."
- Value- Information that people are putting away for later use. Any collection of

bytes can be used as an object's value. The maximum capacity of a single object is 5 terabytes.

- Metadata- User-managed data about the object, represented as a list of name-value pairs. User-defined metadata is a type of metadata that can be attached to objects in Amazon S3. In addition to the object data, Amazon S3 additionally stores system-metadata about the item.
- Subresource-Amazon S3's subresource method is used for storing data about individual objects. Subresources are always linked to another entity, such as an object or a bucket, due to their subordinate status.
- Access control information- When storing data in Amazon S3, one has the ability to restrict access to the objects. Access control lists (ACLs) and bucket policies are two examples of resource-based access control that Amazon S3 enables.

All of our Amazon S3 data (buckets, objects, *etc.*) is secured and encrypted. In order for people to use these facilities, we must give them express permission to do so.

Getting started with S3

Let's get started with Amazon S3 and make our own buckets so that we may store as much or as little data as we like.

1. To do this, go to the S3 homepage. This is shown in Fig. (**1**).

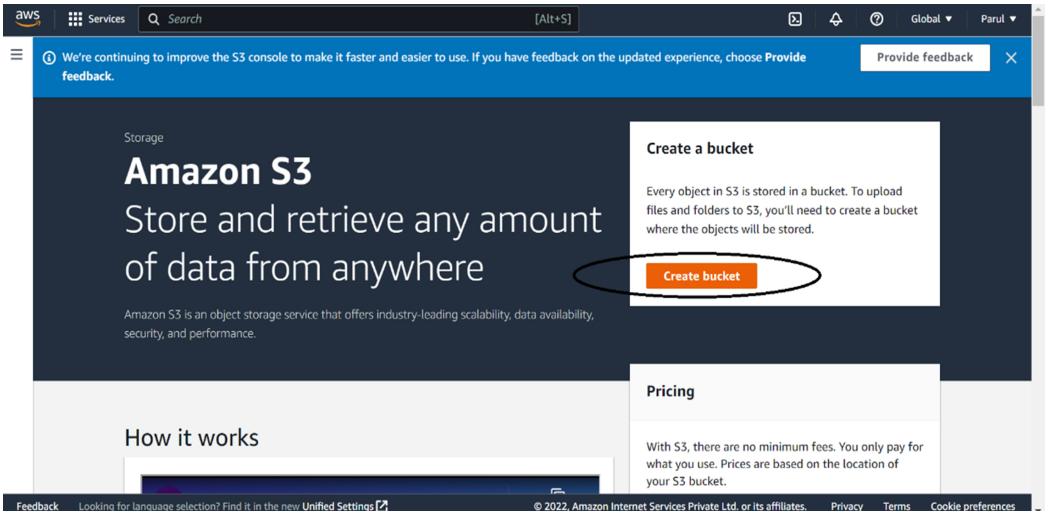


Fig. (1). Beginning with creation of S3 bucket.

CHAPTER 7

AWS Databases and Analytics

Abstract: This chapter provides an in-depth exploration of various databases available within AWS. It focuses on two key databases, namely Amazon RDS and Amazon Aurora, detailing their architecture and functionality. The chapter includes a comprehensive guide to creating a new RDS database, covering all necessary details. Additionally, the chapter introduces a section dedicated to data analytics in AWS. It discusses essential components such as Glue, Redshift, and Quicksight, highlighting their significance and use cases within the AWS ecosystem. By delving into the intricacies of AWS databases and data analytics tools, this chapter equips readers with the knowledge required to make informed decisions and effectively leverage these services. It serves as a valuable resource for understanding the architecture, creation, and practical applications of Amazon RDS, and Amazon Aurora, as well as data analytics tools like Glue, Redshift, and Quicksight within AWS.

Keywords: Amazon RDS, Amazon aurora, Architecture, Databases, Data analytics, Glue, Redshift, Quicksight.

INTRODUCTION

In order for an application to work correctly, the data generated by users, devices, and the app itself must all have a safe and secure place to live. Databases are essential back-end systems that are used to store data, manage that data, update that data, and analyze that data for a broad variety of applications, ranging from locally-based back-office systems to consumer web apps and mobile apps with a global reach.

In computer parlance, “database” refers to any organized grouping of records that can be accessed in a structured manner. Words, figures, pictures, movies, and files are just a few examples of the information that may be stored in this format. A DBMS is a program used to handle large amounts of data, including its storage, retrieval, and modification. When discussing computer systems, the term “database” may also refer to a database management system (DBMS), the database itself, or an application that interacts with databases.

Each and every enterprise has to have access to a dependable database. Databases are essential for businesses because they assist with internal procedures and allow

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

them to monitor their interactions with customers and suppliers. In addition, it stores administrative data as well as information that is more specialized, such as engineering or economic models. A few examples of technological advancements are online catalogues, reservation systems for holidays, and inventory management software for items. There are numerous different applications for databases; however, some of the most significant ones are efficient scaling, data integrity, data security, and data analytics.

MODERN DATABASES

Modern databases have developed the ability to expand in both vertical and horizontal directions. They have the capacity to store massive amounts of data in the cloud and give sophisticated user interfaces for data analytics tools like machine learning [86].

Cloud Database

In most cases, a cloud database will be hosted by a cloud computing service. Users may either host their own databases in the cloud or pay for access to a cloud database provider's infrastructure. Both the SQL and NoSQL data models are supported by cloud databases.

Graph Database

When it comes to organizing data, graph databases are significant because they highlight connections between entries. They are designed for the sole purpose of storing and navigating connection. There are two main parts to a graph database: nodes and edges. Information is organized into nodes, which hold the data itself, and edges, which hold the connections between nodes. Every single edge has a beginning and an ending node, as well as a type and a direction. It may be used to define ownership, responsibilities, and behaviors between parents and children. The amount and types of connections between nodes are unlimited.

In-memory Database

The in-memory database, in contrast to the vast majority of databases that are saved on removable media, is kept entirely inside the computer's main memory. In most cases, however, it is also backed up by electronic data storage. Faster performance is achieved by using in-memory databases as opposed to disc databases. They have widespread use in systems where a fast reaction time is essential, such as in the telecommunications industry.

AWS CLOUD DATABASES

The Amazon Web Services (AWS) Cloud Databases product family provides users with a broad range of adaptable database options. AWS databases allow everything from the creation of servers and the application of patches to the configuration of databases and the creation of backups. As a result, programmers are free to focus on the creation of applications while Amazon Web Services (AWS) monitors the databases and automatically tackles scalability issues. The following are some examples of typical database services offered by AWS:

- Amazon Relational Database Service (RDS) is a managed service that makes the process of building, extending, and managing a relational database in the cloud much more streamlined.
- The relational database hosted in the cloud by Amazon Aurora is an efficient tool. This fully managed solution may relieve users of a number of time-consuming responsibilities, including provisioning, patching, backup and recovery, as well as failure detection and repair.
- The graph-oriented Amazon Neptune database is capable of processing more than one hundred thousand queries per second.
- Solution for data warehouses hosted in the cloud Amazon Redshift is capable of managing petabytes of data and is totally under Amazon's control.
- In-memory NoSQL database that is completely controlled, scalable, and secure; it is offered by Amazon as ElastiCache. It is particularly useful for mobile apps, video games, online businesses, and other applications that depend on data that is routinely available and stored in memory.

Amazon Relational Database Service

Utilizing Amazon's Relational Database Service (Amazon RDS) makes it very simple to create, maintain, and expand a relational database that is hosted in the cloud. In addition to providing scalable capacity at a cheap cost, it automates onerous administrative activities such as the provisioning of hardware, the configuration of databases, the application of patches, and the creation of backups. Because of this, we will be able to put more of our attention on enhancing the compatibility, availability, performance, and safety of our apps [87, 88].

The Amazon Relational Database Service, often known as RDS, is an all-encompassing database management solution that is compatible with a wide variety of database engines. Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server are all examples of these types of databases. By using the AWS Database Migration Service, it is possible to duplicate any

CHAPTER 8

Compute Services in AWS

Abstract: This chapter provides a comprehensive study of container services, specifically focusing on Docker. It includes a detailed comparison between Docker implementation and virtual machines. The chapter offers an overview of various container services provided by AWS, such as Amazon Elastic Container Service (ECS), AWS Fargate, Amazon Elastic Container Registry, AWS Copilot, AWS Batch, and Kubernetes. Additionally, the chapter explores Lambda, a serverless service offered by AWS. It includes practical guidance on creating a lambda function. The chapter also touches upon Lightsail, catering to readers with less experience in AWS. By covering a wide range of topics, from Docker and container services to Lambda and Lightsail, this chapter equips readers with the knowledge and practical skills necessary to work effectively within the AWS ecosystem. It serves as a valuable resource for understanding and utilizing container services, serverless computing, and AWS services tailored for different user levels.

Keywords: Amazon elastic container service (ECS), AWS fargate, Amazon elastic container registry, Container services, Docker, Kubernetes, Lambda, Virtual machine.

INTRODUCTION

The source code of the application, together with its settings and any dependencies, may often be encapsulated into a single package using containers, which is a popular way. Containers allow for deployments to be carried out quickly, consistently, and regardless of the surrounding environment. This is made possible by the fact that containers share a single server's operating system despite being functionally independent processes that do not share any other system resources.

Developers have a powerful tool at their disposal in containers for bundling and delivering software. Because of their compact size and standardized, upgradable software environment, they are appropriate for deployment in almost any setting. Among the most common applications for containers today are the construction and deployment of microservices, the execution of batch processes for machine learning applications, and the migration of on-premises systems to cloud storage.

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

Docker is a platform that may be used for application creation, testing, and deployment. Docker is a system that packages distributed programs into portable containers that comprise the application's source code, dependencies, and runtime environment. This automates the process of distributing and executing distributed applications, which was previously a manual operation.

WORKING OF DOCKERS AND VIRTUAL MACHINES

In the current climate, organizations desire to go through digital transformations but are limited in their efforts since there are so many application, cloud, and on-premises infrastructure alternatives available [92]. Docker is a container platform that assists enterprises in overcoming this challenge by merging old applications and microservices created for Windows, Linux, and mainframe systems into an automated and secure distribution network.

Built on top of the virtualization technology docker, Docker is a container-based software development, deployment, and management tool. Docker was named after its underlying technology. A container is a tiny, self-contained executable package of a program that contains all of its essential components, such as libraries, configuration files, and dependencies. A container may be thought of as an encapsulation of the program's dependencies.

To put it more simply, the container serves as the environment throughout the software development life cycle of the program. This ensures that the application continues to function in the same manner regardless of where it is deployed or which machine it is running on. Due to the enhanced level of protection that they provide, several containers may run simultaneously on the same host. This is made possible by the isolation that containers provide. In addition, since they don't need a hypervisor, containers don't contribute much more weight to the system than they already do. In contrast to a hypervisor, which functions as a guest operating system similar to that of VMWare or VirtualBox, containers are built right into the kernel of the machine that works as the host.

On the other hand, a Virtual Machine (VM) is constructed such that it can carry out actions that cannot be executed directly on the host environment because they would be too risky. Because virtual machines, also known as VMs, are entirely isolated from the rest of the system, it is impossible for the host computer to be harmed by any programs that are executing on a VM. As a result, virtual computers are used for operations such as retrieving data that has been infected by viruses and testing operating systems.

The following is an example of a virtual machine: A virtual machine, often known as a VM, is an instance of a computer program or operating system that runs in

the form of an image inside another program or system known as the "host." Because a virtual machine may operate independently of the operating system on which it is hosted, it is well-suited for usage in a variety of contexts, including the production of system backups, the execution of software and applications, and beta testing.

It is feasible for a single host to act as a host for several virtual machines at the same time. A logfile, a file for establishing the NVRAM, a file for the virtual disc, and a configuration file are some of the crucial files that are included in a virtual machine. Another field that may reap significant advantages from using virtual machines is server virtualization. A single physical server may be converted into a large number of virtual servers so that each one can independently run its own operating system without affecting the performance of the others. Every single virtual machine ships with its own virtual central processing units, RAM, network adapters, hard discs, and everything else that it would need in order to run properly. The comparison between the working of the docker and the virtual machine can be seen in Fig. (**1**) below:

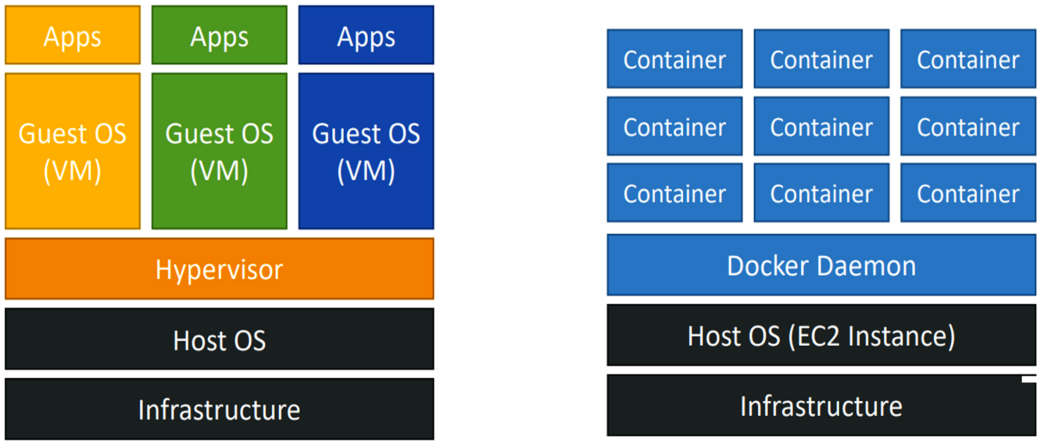


Fig. (1). Comparison between working of docker and virtual machine.

Run Docker on AWS

When it comes to Docker, AWS provides support for both open-source and commercial versions of the software. Amazon Elastic Container Solution (ECS), sometimes known as Amazon ECS, is a container management service that offers excellent scalability and performance, and it is only one of the many alternatives available for hosting containers on AWS. Customers are able to migrate their containerized applications from their local Docker environment to Amazon Elastic Container Service in an efficient and uncomplicated manner (ECS). AWS

CHAPTER 9

Cloud Integrations

Abstract: This chapter delves into the application integration services offered by AWS, which proves particularly beneficial for microservices, distributed systems, and serverless applications. These services facilitate seamless communication between loosely related components. The chapter provides an overview of various AWS integration services, emphasizing their importance in integration processes. Key topics covered in this chapter include SQS (Simple Queue Service), SNS (Simple Notification Service), Kinesis, Pinpoint, and Amazon MQ. Furthermore, the chapter offers practical guidance through hands-on illustrations focused on SQS, allowing readers to gain valuable experience in utilizing these services. By exploring these AWS application integration services, readers can enhance the efficiency and effectiveness of their integration processes. This chapter serves as a comprehensive guide for understanding and utilizing services like SQS, SNS, Kinesis, Pinpoint, and Amazon MQ, enabling seamless communication and integration within AWS environments.

Keywords: Amazon MQ, Distributed systems, Microservices, Pinpoint, Serverless, SQS, SNS, Kinesis.

INTRODUCTION

Microservices, distributed systems, and serverless applications might all stand to gain from the application integration services offered by AWS. These services provide a way for components that are only weakly connected to one another to communicate with one another. It is possible that decoupling applications of any size may decrease the impact of changes, making it easier to update and allowing for speedier feature releases, all without the need to rethink the whole architecture.

- **Increase the agility -** By integrating the apps with the assistance of application integration services, businesses may save the time and effort required to write custom code to ensure compatibility. As a consequence of this, the likelihood that the microservices and functions would include duplicated code is decreased.
- **Independent scaling and failure -** The application may continue to be compatible with application integration services even if they are decoupled from one another; nevertheless, the compatibility of the app with those services will

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

- not be impacted by the failure of one service or a sudden rise in the workload of that service.
- **Concentrate on innovation. -** When we utilize the application integration services provided by AWS, customers are freed from the day-to-day operations of the business, which enables businesses to devote more of their time and effort to the development of innovative and helpful features. The manual provisioning of servers, administration, and updating of security software are no longer necessary thanks to automatic scaling.
- **Communicate with assurance -** Application integration messaging services make advantage of cross-availability zone message storage to ensure the availability and lifespan of messages regardless of the throughput of the service.

AWS APPLICATION INTEGRATION SERVICES

AWS's application integration services are divided into five distinct categories. These include API administration, event bus, messaging, API integration with no coding, and workflows.

1. API management is the process of designing, publishing, managing, monitoring, and protecting API at any size for online applications and workloads. It may also provide an API that enables safe access, modification, and combining of data from several data sources.

2. Event bus facilitates the development of an event-driven architecture that links application data from our own SAAS and AWS services.

3. Amazon's simple notification service is responsible for dependable, high-throughput SMS, email, and mobile push notifications, which are an integral aspect of the application integration service. As part from messaging, message queues are responsible for transmitting, storing, and receiving messages between any number of application components. Under Amazon MQ, a message broker that facilitates migration and enables hybrid design is a crucial component.

4. Amazon app flow under no code API integration automates the flow of data between saas apps and AWS services at any scale without requiring the use of code.

5. The workflow procedure consists of AWS step functions and Amazon workflow management for Apache airflow. This is responsible for combining various services into a serverless process in order to rapidly create and update applications, as well as operating Apache airflow at scale without destroying our infrastructure for managing services.

In this chapter, we are going to focus on the messaging part of the AWS Cloud integration.

AWS MESSAGING

Applications and endpoints developed in a broad range of programming languages and operating systems are able to communicate data and coordinate their activities by using the messaging services provided by AWs [94]. It is possible to use the messaging services provided by AWS in order to send and receive information with regard to cloud programs. The underlying infrastructure is automatically set up for high availability and message durability to guarantee the dependability of our applications. This ensures that our applications will always work as expected.

Amazon MQ

Message brokers make it possible for data to go in both directions across several software systems, regardless of the programming language used to create them or the computer environment in which they are deployed. Amazon MQ is a managed message broker service that may be used if we're working with Apache ActiveMQ or RabbitMQ and want to make it easier to set up and maintain the message brokers on AWS [95].

Amazon MQ makes it easy to setup and maintain message brokers on the Amazon Cloud by way of Apache ActiveMQ and RabbitMQ by offering a managed service. Users have access to all of the most common messaging protocols and application programming interfaces (APIs), including JMS, NMS, AMQP 1.0 and 0.9.1, STOMP, MQTT, and WebSocket. Users also have access to the management interface for ActiveMQ and RabbitMQ. As long as the applications in question comply with the standards established by these brokers, it will not be necessary to rewrite them in order to migrate them over to Amazon MQ for processing. Use cases of Amazon MQ can be listed as follows:

1. Minimal-Delay Event Communication- Make sure thousands of events have minimal latency so that apps may get the most up-to-date data.

2. Move with adaptable configurations-Active/standby, a broker network, and cluster architectures are all supported throughout the transition from on-premises to Amazon MQ.

3. Lambda invocations-The Amazon MQ message broker may be polled using Lambda functions, allowing applications to be integrated.

CHAPTER 10

Cloud Monitoring

Abstract: This chapter provides a comprehensive exploration of Amazon Web Services (AWS) monitoring. It focuses on AWS CloudWatch and associated logs, covering various aspects in detail. The chapter includes step-by-step instructions for setting up CloudWatch alarms, working with CloudWatch logs, and monitoring CloudTrail. Furthermore, the chapter delves into the AWS health check feature, providing a deeper understanding of its significance. It highlights the importance of both private health checks and the Service Health Dashboard, emphasizing their role in ensuring system reliability and performance. By studying AWS monitoring tools and features, readers gain insights into effectively monitoring their AWS resources and applications. This chapter serves as a valuable resource for understanding and implementing CloudWatch, CloudTrail, and AWS health checks, enabling users to maintain a proactive and robust monitoring approach within the AWS environment.

Keywords: Cloud watch, Cloud trail, Health check, Logs, Monitoring, Private health checks, Service health dashboard.

INTRODUCTION

Through the use of Apps Manager's cloud monitoring, customers will be able to get a profound comprehension of the state of both the hybrid and multi-cloud infrastructure as well as the apps that are operating on it. Through the use of cloud monitoring, web services and applications that are hosted in the cloud by companies such as Amazon Web Services (AWS) are inspected for uptime, health, and performance.

The process of monitoring and regulating the inner workings of an information technology system that is housed in the cloud is referred to as monitoring in the cloud. In either a manual or an automated management process, websites, servers, applications, and other cloud infrastructures are monitored for uptime and performance. This monitoring may be done manually. The early identification of potential vulnerabilities is made possible by continuous monitoring of server response times and speeds, as well as available resources. Cloud monitoring is beneficial in the following ways:

Parul Dubey, Arvind Kumar Tiwari & Rohit Raja

1. The straightforward and efficient way of scaling up to handle increasing activity levels may be beneficial to businesses of varying sizes.

2. The required specialized equipment is owned by the hosting business, and it is also managed by that firm.

3. Because these solutions can be used across mobile devices, desktop computers, and tablets, the company is able to monitor app activity regardless of where it is located.

4. Installation is a breeze and takes very little time since all of the prerequisites, including the framework and settings, are already in place.

5. As a result of the fact that these resources are not a part of the company's servers and workstations, the likelihood of the system experiencing downtime as a result of a local problem is significantly reduced.

6. Services that are provided on a subscription basis might end up saving customers money in the long term.

AMAZON CLOUDWATCH

Amazon CloudWatch is a service that monitors the Amazon Web Services (AWS) resources as well as the applications that run on Amazon Web Services. With the assistance of Amazon CloudWatch, metrics, log files, and alarms can all be grouped together and watched individually. Amazon CloudWatch enables users to monitor their Amazon Elastic Compute Cloud (EC2) instances, Amazon DynamoDB tables, Amazon Relational Database Service (RDS) database instances, as well as application log files and any custom metrics that one’s apps and services generate. Amazon CloudWatch gives users visibility into metrics relating to the whole system, such as the consumption of resources, the performance of applications, and the health of operations [102, 104]. We may use this information to make modifications and ensure that the app continues to operate in a reliable manner.

Amazon CloudWatch is responsible for the collection of logs, metrics, and events in real time. These are then shown in interactive dashboards with the purpose of making the maintenance of infrastructure and applications simpler.

Access to Amazon CloudWatch may be gained *via* the AWS Management Console, AWS Software Development Kits (SDKs), the Amazon Elastic Compute Cloud (EC2), and the command line interface.

Amazon CloudWatch enables users to regulate which CloudWatch activities each user in the AWS Account is authorized to carry out. This is made possible by Amazon CloudWatch's interoperability with the Identity and Access Management (IAM) service provided by AWS. One may, for instance, design an IAM policy in order to limit access to Get Metric Statistics to a certain group of the organization's workers. It is possible for a third party to get information about the cloud services by following this process.

It is not possible to use IAM to limit who may see the data in CloudWatch for certain resources. For example, it is not possible to provide someone access to the data stored in CloudWatch for only a small subset of instances or a single Load Balancer. When we use CloudWatch, all of the cloud resources that users access are safeguarded by the rights that are granted by IAM. In addition to this, the Amazon CloudWatch CLI tools do not support IAM roles in their implementation.

CLOUDWATCH LOGS

When monitoring and resolving issues, customers may make use of their existing system, application, and custom log files in conjunction with Amazon CloudWatch Logs.

CloudWatch Logs gives the ability to do near real-time searches of the logs for certain keywords, values, and patterns. One may generate an alert based on the number of failures that are recorded in the system logs, or can monitor the latency of web requests that are made in the application logs [104, 105]. Examining the data from the first log file might help determine the source of the issue after it has been identified. Because log data can be stored and retrieved continually from highly long-lasting storage at a cheap cost, there is no way for the hard drives to ever run out of storage capacity.

Users will have a more in-depth understanding of how their systems and applications are operating as a result of using CloudWatch Logs since it allows them to monitor and preserve all logs for later investigation.

One may maintain a real-time watch on their applications and servers by using CloudWatch Logs in the way that is most convenient for them. We have the option of configuring CloudWatch Logs to send users an alert in a variety of situations, such as when the number of errors recorded in the application logs reaches a predetermined threshold. Because it makes use of existing log data, CloudWatch Logs allows monitoring without having to make any changes to the code.

REFERENCES

[1] Available From: https://www.latimes.com/archives

[2] E. Messmer, "Security of virtualization, cloud computing divides IT and security pros.Network World. Network World", Available From: https://www.networkworld.com/article/2244954/security-o--virtualization--cloud-computing-divides-it-and-security-pros.html

[3] B. Butler, "What is hybrid cloud computing? The benefits of mixing private and public cloud services. Network World. Network World", Available From: https://www.networkworld.com/article/3233132/what-is-hybrid-cloud-computing.html

[4] M.D. Ryan, "Cloud computing privacy concerns on our doorstep. cloud computing privacy concerns on our doorstep", *Commun. ACM.,* vol. 54, no. 1, pp. 36-38, 2011. Available From: https://cacm.acm.org/magazines/2011/1/103200-cloud-computing-privacy-concer-s-on-our-doorstep/fulltext
[http://dx.doi.org/10.1145/1866739.1866751]

[5] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "CloudID: Trustworthy cloud-based and cross-enterprise biometric identification", *Expert Syst. Appl.,* vol. 42, no. 21, pp. 7905-7916, 2015.
[http://dx.doi.org/10.1016/j.eswa.2015.06.025]

[6] H. Salcedo, "Google drive, dropbox, box and icloud reach the top 5 cloud storage security breaches list. google drive, dropbox, box and icloud reach the top 5 cloud storage security breaches list", 2014. Available From: https://web.archive.org/web/20151123032912/https://psg.hitachi-solutions.com/credeon/blog/google-drive-dropbox-box-and-icloud-reach-the-t-p-5-cloud-storage-security-breaches-list (Accessed on 2014 November 20).

[7] Available From: https://en.wikipedia.org/wiki/Cloud_computing (Accessed on 2021 March 1).

[8] Available From: https://en.wikipedia.org/wiki/Cloud_computing_architecture (Accessed on 2012 May 27)

[9] Available From: https://www.pcmag.com/encyclopedia/term/saas

[10] D. Athow, "Hybrid cloud: Is it right for your business", *TechRadar.,* 2014. Available From: https://www.techradar.com/news/internet/cloud-services/hybrid-cloud-is-it-right-f-r-your-business-1261343 (Accessed on 2014 August 24).

[11] Available From: https://www.techtarget.com/searchcloudcomputing/definition/cloud-bursting (Accessed on 2021 June 1).

[12] Available From: https://www.techtarget.com/searchcloudcomputing/definition/hybrid-cloud (Accessed on 2021 September 1).

[13] Available From: https://blogs.gartner.com/thomas_bittman/2012/09/24/mind-the-gap-here-c-mes-hybrid-cloud/ (Accessed on 2012 September 24)

[14] C. Miyachi, "What is 'Cloud'? It is time to update the NIST definition?," IEEE Cloud Computing, vol. 5, no. 3, pp. 6–11, May 2018.
[http://dx.doi.org/10.1109/mcc.2018.032591611]

[15] Available From: https://www.scribd.com/document/64699897/Introduction-to-Cloud-Comput-ng-Business-and-Technology

[16] M. Vizard, "How Cloudbursting "Rightsizes" the Data Center", *Dice Insights,* 2012. Available From: https://insights.dice.com/2012/06/21/how-cloudbursting-rightsizes-the-data-center/ (Accessed on 2012 June 22).

[17] Y. Sun, J. Zhang, Y. Xiong, and G. Zhu, "Data security and privacy in cloud computing", *Int. J. Distrib. Sens. Netw.,* vol. 10, no. 7, p. 190903, 2014.
[http://dx.doi.org/10.1155/2014/190903]

[18] Y. Jiang, K. Zhang, Y. Qian, and R. Q. Hu, "Efficient and Privacy-preserving Distributed Learning in

Cloud-Edge Computing Systems", *Proceedings of the 3rd ACM Workshop on Wireless Security and Machine Learning,* 2021. Available From: https://dl.acm.org/doi/abs/10.1145/3468218.3469044 (Accessed 2021 June 1)

[19] J. Furrier, "Exclusive Profile: Andy Jassy of Amazon Web Service (AWS) And His Trillion Dollar Cloud Ambition", *Forbes,* 2015. Available From: https://www.forbes.com/sites/siliconangle/2015/01/28/andy-jassy-aws-trillion-dollar-cloud-ambition/?sh=2afd0029321e(Accessed on 2015 January 28)

[20] R. Miller, "How AWS came to be | TechCrunch", *How AWS came to be | TechCrunch,* 2016. Available From: https://techcrunch.com/2016/07/02/andy-jassys-brief-history-of-the-genesis-of-aws/

[21] K. McLaughlin, "Andy Jassy: Amazon's $6 Billion Man | CRN", *CRN,* 2015. Available From: https://archive.ph/20210203122831/https://www.crn.com/news/cloud/300077657/andy-jass--amazons-6-billion-man.htm

[22] Available From: https://www.youtube.com/watch?v=d2dyGDqrXLo (Accessed on 2013 October 21)

[23] Available From: https://queue.acm.org/detail.cfm?id=3352692

[24] Available From: https://books.google.co.in/books?id=yG3PAK6ZOucC&redir_esc=y

[25] Available From: https://www.youtube.com/watch?v=6PiyzyWXiIk&t=300s (Accessed on 2013, September 6)

[26] Available From: https://www.networkworld.com/article/2329864/linuxworld--amazon-s-two--aces-present-it-challenge.html

[27] Available From: https://www.carbonblack.com/press-releases/former-amazon-ciso-tom-killalea-joins-carbon-black-board/

[28] Available From: https://press.aboutamazon.com/news-releases/news-release-details/amazon-om-launches-web-services

[29] Available From: https://www.computerweekly.com/news/2240054229/Amazon-lauds-Lin-x-infrastructure

[30] Available From: https://www.allthingsdistributed.com/2019/08/modern-applications-at-aws.html (Accessed on 2019 August 28)

[31] Available From: https://web.archive.org/web/20170802040741/https://www.macfound.org/press/semifinalist-perspectives/what-i-learned-jeff-bezos-aka-how-bring-millions-books-billions-people/

[32] Available From: http://blog.b3k.us/2009/01/25/ec2-origins.html (Accessed on 2009 January 25)

[33] Available From: http://www.businessinsider.com/amazons-game-changing-cloud-was-built-by-some-guys-in-south-africa-2012-3

[34] Available From: https://press.aboutamazon.com/news-releases/news-release-details/amazon-web-services-launches-amazon-s3-simple-storage-service

[35] Available From: https://aws.amazon.com/blogs/aws/amazon_ec2_beta/ (Acessed on 2006 August 25).

[36] Available From: https://www.computerworld.com/article/2549084/computing-in-the-cloud.html

[37] Available From: https://archive.fortune.com/magazines/business2/business2_archive/2006/10/01/8387086/index.htm

[38] N. H. Prickett, "Amazon CTO on Past, Present, Future of S3 - The Next Platform," The Next Platform - In-depth coverage of high-end computing at large enterprises, supercomputing centers, hyperscale data centers, and public clouds", Available From: https://www.nextplatform.com/2021/04/22/amazon-cto-on-past-present-future-of-s3/, 2021.

[39] Available From: https://www.slideshare.net/AmazonWebServices/2011-aws-tour-australia-cloing-keynote-how-amazoncom-migrated-to-aws-by-jon-jenkins (Accessed on 2001 March 1)

[40] Available From: https://books.google.co.in/books?id=jgn5DwAAQBAJ&redir_esc=y

[41] J. O'Dell, "The first one's free, kid.' Amazon launches AWS Activate to get startups hooked", *VentureBeat,* 2013. Available From: https://venturebeat.com/2013/10/11/the-first-ones-free-kid-amazon-launches-aws-activate-to-get-startups-hooked/ (Accessed on 2013 October 11)

[42] "Announcing the launch of the aws partner network (apn) blog | amazon web services. amazon web services;", Available From: https://aws.amazon.com/blogs/apn/apn-blog-launch/ (Accessed on 2014, November 21)

[43] Available From: https://aws.amazon.com/iot/partner-solutions/edgeiq-orchestration/

[44] Available From: https://www.bbc.co.uk/news/business-32442268

[45] C. Nast and @wired,, "Get Used to Amazon Being a Profitable Company, WIRED", Available From: https://www.wired.com/2015/10/get-used-to-amazon-being-a-profitable-company/ (Accessed on 2015 October 22), 2015.

[46] "Amazon's earnings soar as its hardware takes the spotlight: The Verge", Available From: https://www.theverge.com/2016/4/28/11530336/amazon-q1-first-quarter-2016-earnings (Accessed on 2016, April 28)

[47] Available From: https://finance.yahoo.com/news/why-amazon-stock-is-up-42-percent--n-just-3-months-web-services-and-profit-bility-181247737.html?guccounter=1&guce_referrer=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnLw&guce_referrer_sig=AQAAAJgDmsT2XyEGQu--DM4ap93fVU2_dkCQ1kn_e8yL8413swI4pLuXYp5p0KmJ0lzG0BgTVSoKF4S8bvwxdoeH9HAa9-lRHcfYKXCCUUociLBoaTQ6ijpUaW2BqkDToiS4opWY5PqEl50QOWFCE5ACRWBYm2NFB58j0Q10X8X0u1m (Accessed on 2016 May 24)

[48] Available From: https://siliconangle.com/2020/11/30/exclusive-aws-chief-andy-jassy-wakeu--call-cloud-adoption/ (Accessed on 2020 November 30)

[49] R. Miller, "Amazon launches autoscaling service on AWS | TechCrunch," Amazon launches autoscaling service on AWS | TechCrunch,2018, Available From: https://techcrunch.com/2018/01/17/amazon-launches-autoscaling-service-on-aws/

[50] "New AWS Auto Scaling: Unified Scaling For Your Cloud Applications | Amazon Web Services", Available From: https://aws.amazon.com/blogs/aws/aws-auto-scaling-unified-scaling-for-you--cloud-applications/(Accessed on 2018 January 17)

[51] F. Lardinois, "AWS launches Arm-based servers for EC2 | TechCrunch," AWS launches Arm-based servers for EC2 | TechCrunch, 2018., Available From: https://techcrunch.com/2018/11/27/aws-launches-arm-based-servers-for-ec2/

[52] R. Miller, "AWS launches a base station for satellites as a service | TechCrunch," AWS launches a base station for satellites as a service | TechCrunch, 2018., Available From: https://techcrunch.com/2018/11/27/aws-launches-a-base-station-for-satellites-as-a-service/

[53] D. Sparks, "Amazon's Record 2019 in 7 Metrics | The Motley Fool," The Motley Fool, 2020., Available From: https://www.fool.com/investing/2020/02/06/amazons-record-2019-in-7-metrics.aspx (Accessed on 2020 February 6)

[54] Z. Alomari, M. AL-Oudat, and S. Alkhushayni, "Pricing the cloud based on multi-attribute auction mechanism", *Cluster Comput.,* pp. 1-26, 2023. [http://dx.doi.org/10.1007/s10586-023-03975-2]

[55] I. Indu, P. R. Anand, and V. Bhaskar, "Identity and access management in cloud environment: Mechanisms and challenges. Engineering science and technology", *Eng. Sci. Tech. Int. J.,* vol. 21, no. 4, pp. 574-588, 2018.

[56] Available From: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

[57] Available From: https://docs.aws.amazon.com/IAM/latest/UserGuide/id.html

[58] L. Qian, Z. Luo, Y. Du, and L. Guo, "Cloud Computing: An Overview", *Lecture Notes in Computer Science,* pp. 626-631, 2009.
[http://dx.doi.org/10.1007/978-3-642-10665-1_63]

[59] "Security in IAM and AWS STS - AWS Identity and Access Management (n.d.). Security in IAM and AWS STS - AWS Identity and Access Management. ", Available From: https://docs.aws.amazon.com/IAM/latest/UserGuide/security.html

[60] Available From: https://aws.amazon.com/ec2/instance-types/

[61] Q. Li, Q. Hao, L. Xiao, and Z. Li, "Adaptive management of virtualized resources in cloud computing using feedback control", *2009 First Int. Conf. Inform. Sci. Eng.* Nanjing, China pp. 99-102, 2009.
[http://dx.doi.org/10.1109/ICISE.2009.211]

[62] "AWS Product and Service Pricing | Amazon Web Services. (n.d.). Amazon Web Services, Inc, 2009", Available From: https://aws.amazon.com/pricing/

[63] Available From: https://aws.amazon.com/ec2/pricing/on-demand/

[64] Available From: https://www.geeksforgeeks.org/pricing-models-for-amazon-ec2/ (Accessed on 2019, July 11).

[65] Available From: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Storage.html

[66] Available From: https://www.geeksforgeeks.org/what-is-aws-ec2-instance-storage/ (Accessed on 2020 July 24)

[67] J. Baron, and S. Kotecha, *Storage options in the aws cloud. Amazon web services.*. Tech. Rep: Washington, DC, 2013.

[68] S. Mathew, and J. Varia, "Overview of amazon web services", *Amazon Whitepapers,* vol. 105, pp. 1-22, 2014.

[69] P. Dubey, and A.K. Tiwari, "Cloud integration in the educational sector using amazon web services", *2023 Third Int. Conf. Adv. Elect. Comp. Commun. Sustain. Technol. (ICAECT)* Bhilai, India pp. 1-5, 2003.
[http://dx.doi.org/10.1109/ICAECT57570.2023.10117739]

[70] B. Gupta, P. Mittal, and T. Mufti, "A review on Amazon web service (AWS), Microsoft azure & Google cloud platform (GCP) services", *In Proceedings of the 2nd Int. Conf. ICT. Dig. Smart. Sustain. Develop.* 2021.

[71] K. Kalpana, P. Thrilochani, P. Srinivasu, S. K. Sameer, T. V. Gopinadh, and U. S. Ram, "Dual access for cloud based data storage and sharing in amazon web services", *Int. Res. J. Mod.Eng. Tech. Sci.,* vol. 4, 2022.

[72] P. Dubey, and A. K. Tiwari, "Solution Architecting on Remote Medical Monitoring with AWS Cloud and IoT", *EAI Endorsed Trans. Perv. Heal. Tech.,* vol. 4, no. 14, 2023.
[http://dx.doi.org/10.1007/978-981-99-0377-1_12]

[73] N. Hota, and B.K. Pattanayak, "Cloud computing load balancing using Amazon web service technology", In: *Prog. Adv. Comp. Intell. Eng*. Springer, pp. 661-669, 2021.
[http://dx.doi.org/10.1007/978-981-33-4299-6_54]

[74] R. Kumari, and V.K. Jha, "Performance analysis of load balancing algorithms in Amazon cloud", In: V. Nath, J.K. Mandal, Eds., *Proc Fifth Int. Conf. Microelect. Comp. Commun. Sys*. vol. 748. Springer: Singapore, pp. 401-412, 2021.
[http://dx.doi.org/10.1007/978-981-16-0275-7_33]

[75] "What is Elastic Load Balancing? - Elastic Load Balancing. (n.d.). What Is Elastic Load Balancing? -

Elastic Load Balancing", Available From: https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html

[76] "Elastic Load Balancer in AWS - GeeksforGeeks. (2020, June 25). GeeksforGeeks. Retrieved",

[77] M. Kushwaha, B.L. Raina, and S.N. Singh, "Implementationanalysis of load balancing procedures for cloud computing domain", *Int. Conf. Comp. Commun. Intell. Sys. (ICCCIS)* Greater Noida, India pp.287-292, 2021.
[http://dx.doi.org/10.1109/ICCCIS51004.2021.9397069]

[78] "Load Balancer - Amazon Elastic Load Balancer (ELB) - AWS. (n.d.). Amazon Web Services, Inc", Available From: https://aws.amazon.com/elasticloadbalancing/

[79] P. Dubey, H. Chourasia, and S. Ghode, "A novel approach to degin home automation using IoT applications", *2023 Int. Conf. Adv. Comp. Comp. Technol. (InCACCT)* Gharuan, India pp.801-806, 2023.
[http://dx.doi.org/10.1109/InCACCT57535.2023.10141754]

[80] G.S. Sharvani, "An auto-scaling approach to load balance dynamic workloads for cloud systems", *Turk. J. Comp. Math. Educ.,* vol. 12, no. 11, pp. 515-531, 2021.

[81] "The Amazon Simple Storage Service | All Things Distributed. S3 - The Amazon Simple Storage Service | All Things Distributed", Available From: https://www.allthingsdistributed.com/2006/03/s3.html

[82] Free Cloud Computing Services - AWS Free Tier, "Amazon Web Services", n.d. Available From: https://aws.amazon.com/free/

[83] Available From: https://aws.amazon.com/s3/

[84] Available From: https://aws.amazon.com/s3/faqs/

[85] Available From: https://aws.amazon.com/s3/storage-classes/intelligent-tiering/

[86] Available From: https://aws.amazon.com/s3/storage-classes/glacier/

[87] Available From: https://aws.amazon.com/free/database/

[88] Available From: https://aws.amazon.com/rds/

[89] A. Verbitski, A. Gupta, D. Saha, M. Brahmadesam, K. Gupta, and R. Mittal, "Amazon aurora: Design considerations for high throughput cloud-native relational databases", *Proceedings of the 2017 ACM International Conference on Management of Data,* 2017pp. 1041-1052
[http://dx.doi.org/10.1145/3035918.3056101]

[90] S. Mathew, and J. Varia, "Overview of amazon web services", *Amazon Whitepapers,* vol. 105, pp. 1-22, 2014.

[91] "Amazon DynamoDB – a Fast and Scalable NoSQL Database Service Designed for Internet Scale Applications | All Things Distributed. Amazon DynamoDB – a Fast and Scalable NoSQL Database Service Designed for Internet Scale Applications | All Things Distributed", Available From: https://www.allthingsdistributed.com/2012/01/amazon-dynamodb.html (2012, January 18)

[92] Compute Services - Overview of Amazon Web Services, "(n.d.). Compute Services - Overview of Amazon Web Services", Available From: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/compute-services.html

[93] Available From: https://aws.amazon.com/docker/

[94] Available From: https://aws.amazon.com/fargate/

[95] Available From: https://aws.amazon.com/messaging/

[96] Available From: https://aws.amazon.com/amazon-mq/

[97] Available From: https://aws.amazon.com/sqs/

[98] Available From: https://aws.amazon.com/sns/

[99] Z. Alomari, M.F. Zhani, M. Aloqaily, and O. Bouachir, "On minimizing synchronization cost in nfv-based environments", *2020 16th Int. Conf. Netw. Serv. Manag. (CNSM),* 2020 Izmir, Turkey [http://dx.doi.org/10.23919/CNSM50824.2020.9269121]

[100] Z. Alomari, M.F. Zhani, M. Aloqaily, and O. Bouachir, "Towards optimal synchronization in NFV-based environments", *Int. J. Netw. Manage.,* vol. 33, no. 1, p. e2218, 2023. [http://dx.doi.org/10.1002/nem.2218]

[101] Multichannel Customer Communications – Amazon Pinpoint – Amazon Web Services, "(n.d.). Amazon Web Services, Inc", Available From: https://aws.amazon.com/pinpoint/

[102] Available From: https://aws.amazon.com/kinesis/

[103] A. Nagaraj, "Introduction to sensors in IoT and cloud computing applications", Bentham Science Publishers, 2021. [http://dx.doi.org/10.2174/9789814793591210101]

[104] Application and Infrastructure Monitoring – Amazon CloudWatch – Amazon Web Services, "(n.d.). Amazon Web Services, Inc. Retrieved from", Available From: https://aws.amazon.com/cloudwatch/

[105] Available From: https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html

[106] Available From: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/working_with_metrics.html

[107] Available From: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html

SUBJECT INDEX

A

B

C

D

E

F

H

I

K

L

M

N

O

P

Q

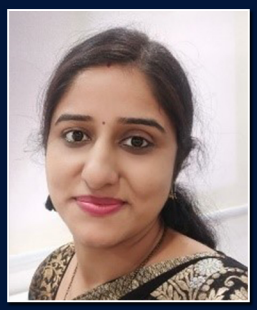
R

S

T

V

W



Parul Dubey

Prof. Parul Dubey is currently working as an assistant professor in AI Department, G H Raisoni College of Engineering, Nagpur, India. She did Masters in computer engineering. She also did a Ph.D. and she is a research scholar in computer science and engineering at C.V. Raman University, Bilaspur, Chhattisgarh, India. She has good academic and research experience in various areas of CSE and IT. She is indulged in research activities which include book chapters, books, conference papers and journal articles as well. She has 12 Indian published patents and 1 Indian granted patent. She has a number of publications which are a part of conferences, Scopus and other journals as well. She is an AWS Certified Cloud Practitioner badge holder, which is a proof for the expertise in AWS. Currently, guiding many engineering students to learn the AWS platform and complete their projects.

Arvind Kumar Tiwari

Prof. Arvind Kumar Tiwari is a highly respected academic and researcher in the field of computer science. He is currently serving as professor and dean of academics at Dr. C. V. Raman University Chhattisgarh, India. He made significant contributions to the field of computer science and has published numerous research papers in prestigious international journals and conferences. In addition to his research work, he is a dedicated teacher and mentor. He has taught several undergraduate and graduate courses in computer science and has guided several Master's students. He has also served on various academic and administrative committees in the university and has been actively involved in organizing conferences and workshops.

Rohit Raja

Prof. Rohit Raja is working as associate professor & head of the IT Department at the Guru Ghasidas, Vishwavidyalaya, Bilaspur (CG). He has done Ph.D. in computer science and engineering in 2016 from C. V. Raman University India. His main research interests include face recognition, signal processing, and networking and data mining. He has successfully filed 27 patents. He has published a number of research papers in various international/national journals (including IEEE, Springer, etc.) and proceedings of the reputed international/ National Conferences (including Springer and IEEE). He received the "Excellence in Research" Award under the Category of HEI Professor by Auropath Global Awards, 2019 and the "Distinguished Research Award "I2OR Awards 2019 during the 4th International Conclave on Interdisciplinary Research for Sustainable Development 2020 (IRSD 2020). He is a member of a professional international society including IEEE, ISTE, CSI, ACM, etc. He is the editor/reviewer of many peer-reviewed and refereed journals, such as IEEE, Springer, Enders Science, etc.