

**Smart Text Extraction**  
**Automated Extraction for Enhanced Insights**

*A report submitted in partial fulfillment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**Information technology**

**By**

**Mohammad Jabir**

**20B91A12B8**

**Under Supervision of Mr. AASHU DEV**

**(Duration: 5th June 2023 to 4th August 2023)**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**S.R.K.R. ENGINEERING COLLEGE**

**(Autonomous)**

**SRKR MARG, CHINNA AMIRAM, BHIMAVARAM-534204, A.P**

**(Recognized by A.I.C.T.E New Delhi) (Accredited by NBA & NAAC)**

**(Affiliated to JNTU, KAKINADA)**

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE  
(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY



**CERTIFICATE**

This is to certify that the Summer Internship Report titled "Smart Text Extraction:Automated Extraction for advanced Insights" is the bonafide work done by Mr. Mohammad Jabir bearing 20B91A12B8 at the end of third year second semester at Blackbuck Engineer Pvt Ltd from 5th June 2023 to 4th August 2023 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science.

**Department Internship Coordinator**

**Dean -T & P Cell**

**Head of the Department**

## **Abstract**

Automated Text Extraction is a cutting-edge technology that revolutionizes the process of extracting textual information from diverse sources. With the exponential growth of digital data in various formats, such as documents, images and multimedia content, there is a growing need to efficiently extract and analyze the textual content embedded within these sources. Traditional manual extraction methods are time-consuming, error-prone, and cannot handle the scale and complexity of modern data.

This project presents an advanced automated text extraction framework that leverages state-of-the-art machine learning and natural language processing techniques. By utilizing a combination of AWS services, including VPC, EC2, AWS Lambda, and AWS Glue, along with the advanced machine learning capabilities of Amazon Textract, this project automates the extraction of valuable text data from various file formats. The extracted data is transformed into machine-readable text and stored in AWS S3, while AWS Glue facilitates further data transformations and analysis. Smarttext Extraction empowers businesses with efficient and accurate information, enabling data-driven decision-making and unlocking hidden insights within documents.

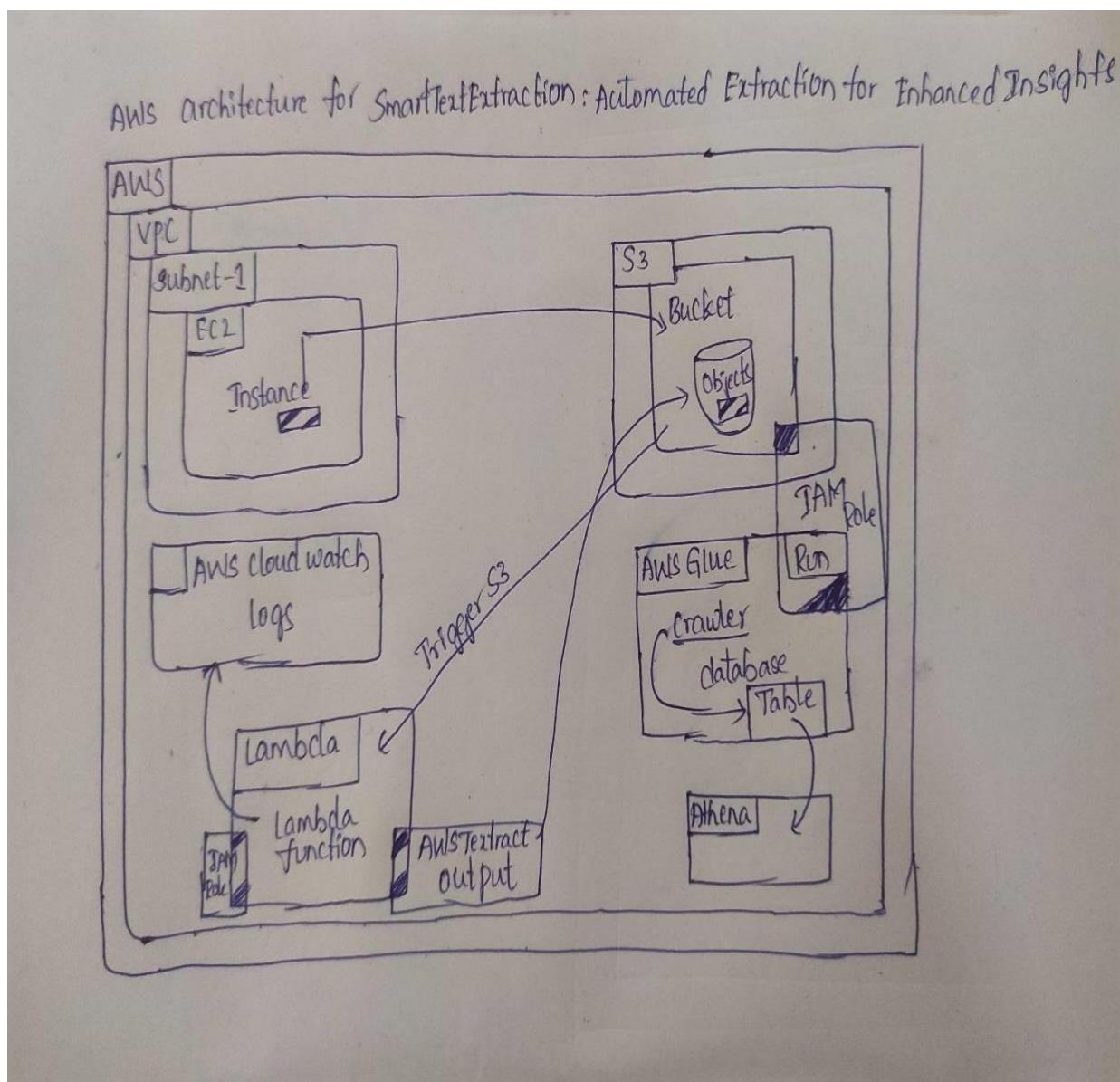
## TABLE OF CONTENTS

Services used.....	1
Rough architecture.....	1
Final architecture.....	2
Description.....	2
Cloud computing.....	3
Cloud computing services.....	4
IaaS(Infrastructure-as-a-Service).....	4
PaaS(Platform-as-a-Service).....	4
SaaS(Software-as-a-Service).....	4
Cloud Service Providers.....	5
Amazon Web Services.....	5
Why AWS?.. ..	6
List of AWS Services Used.....	7
• Amazon EC2.....	8
• Instance types.....	8
• Amazon VPC.....	10
• Amazon S3.....	11
• IAM(Identity and Access Management).....	12
• AWS Lambda.....	13
• Amazon CloudWatch.....	17
• Amazon EBS.....	18
• Amazon Aurora.....	19
• AWS AutoScaling.....	21
Implementation.....	22
Conclusion.....	45

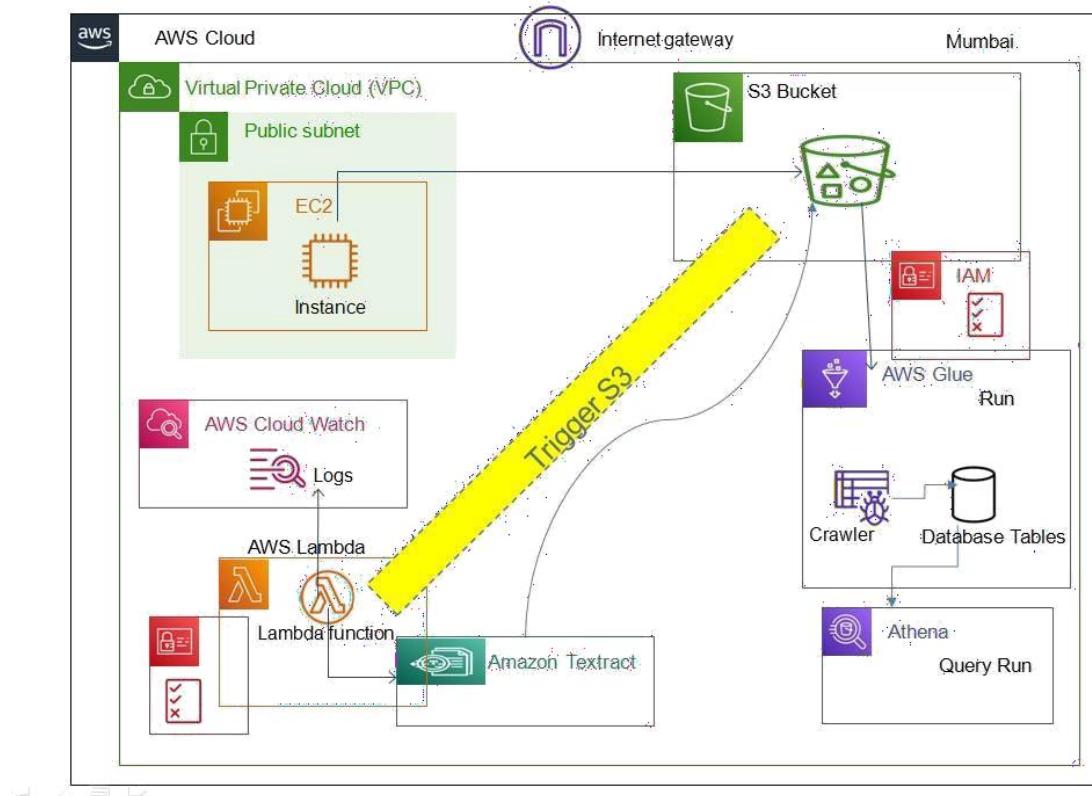
## Services used

- VPC (Virtual Private Cloud)
- EC2 (Elastic Compute Cloud)
- S3 (Simple storage Service)
- AWS Cloud Watch
- IAM (Identity and Access Management)
- AWS Lambda
- Amazon Textract
- AWS Glue
- AWS Athena

## Rough architecture



## Final architecture



## Description

Smarttext Extraction is an innovative project that leverages the power of AWS services to automate the extraction of valuable insights from various types of documents. By utilizing a combination of VPC, EC2, AWS Lambda, and AWS Glue, we have created a seamless workflow for extracting and analyzing text data. At the core of our solution is Amazon Textract, a powerful machine learning service provided by AWS. Amazon Textract enables us to extract text and data from images, scanned documents, and other file formats. By harnessing the capabilities of Amazon Textract, we can efficiently process vast amounts of unstructured data and convert it into a structured format.

The extracted data is stored in AWS S3, providing a secure and scalable storage solution. Images and other file formats are transformed into machine-readable text, allowing for easier analysis and data manipulation. Additionally, the output from Amazon Textract is seamlessly integrated into AWS Glue tables, enabling further data transformations and analysis.

With Smarttext Extraction, businesses and organizations can unlock valuable insights hidden within their documents. By automating the extraction process, we eliminate the need for manual data entry and greatly improve efficiency. This project empowers decision-makers with timely and accurate information, enabling them to make data-driven decisions and gain a competitive edge.

Experience the power of Smarttext extraction and unleash the true potential of your document-based data.

## Cloud Computing

A cloud provider, often known as CSP, manages a remote data center where applications, servers (both real and virtual), data storage, development tools, networking capabilities, and other computing resources are hosted. Cloud computing is the on-demand internet access to these resources. These materials are made available by the CSP in exchange for a monthly subscription fee or usage-based services.

Depending on the cloud services you choose and in comparison, to conventional on-premises IT, cloud computing aids in the following:

- **Lower IT expenses:** By using the cloud, you may outsource some or all of the expenses and labor associated with building, installing, configuring and maintaining your own on-premises infrastructure.
- **Boost agility and time-to-value:** By utilizing the cloud, your company may begin using corporate apps immediately rather of waiting weeks or months for IT to reply to a request, buy and setup supplementary hardware, and install software. Additionally, you may empower some users via the cloud, particularly engineers and data scientists.
- **Boost agility and time-to-value:** By utilizing the cloud, your company may begin using corporate apps immediately rather of waiting weeks or months for IT to reply to a request, buy and set up supplementary hardware, and install software. Additionally, you may empower some users via the cloud, particularly engineers and data scientists.
- **Scale more easily and affordably:** The cloud offers elasticity, allowing you to scale capacity up and down in response to spikes and dips in demand rather of purchasing extra capacity that sits idle during slack times. You may distribute your applications closer to users all over the world by utilizing the worldwide network of your cloud provider.

The technology that makes clouds function is often referred to as "cloud computing." This comprises some type of virtualized IT infrastructure, such as servers, operating systems, networking, and other infrastructure that has been

abstracted using specialized software to allow pooling and dividing without regard to physical hardware limits. One hardware server, for instance, could be split up into several virtual servers.

## **Cloud Computing Services:**

- IaaS (Infrastructure-as-a-Service)
- PaaS (Platform-as-a-Service)
- SaaS (Software-as-a-service)

### **IaaS (Infrastructure-as-a-Service)**

IaaS provides on-demand access to fundamental computing resources—physical and virtual servers, networking, and storage—over the internet on a pay-as-you-go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary on-premises or ‘owned’ infrastructure and for overbuying resources to accommodate periodic spikes in usage.

In contrast to SaaS and PaaS (and even newer PaaS computing models such as containers and serverless), IaaS provides the users with the lowest-level control of computing resources in the cloud. IaaS was the most popular cloud computing model when it emerged in the early 2010s. While it remains the cloud model for many types of workloads, use of SaaS and PaaS is growing at a much faster rate.

### **PaaS (Platform-as-a-service)**

PaaS provides software developers with on-demand platform—hardware, complete software stack, infrastructure, and even development tools—for running, developing, and managing applications without the cost, complexity, and inflexibility of maintaining that platform on-premises.

With PaaS, the cloud provider hosts everything—servers, networks, storage, operating system software, middleware, databases—at their data center. Developers simply pick from a menu to ‘spin up’ servers and environments they need to run, build, test, deploy, maintain, update, and scale applications.

Today, PaaS is often built around *containers*, a virtualized compute model one step removed from virtual servers. Containers virtualize the operating system, enabling developers to package the application with only the operating system services it needs to run on any platform, without modification and without need for middleware.

### **SaaS (Software-as-a-Service)**

SaaS—also known as cloud-based software or cloud applications—is application software that’s hosted in the cloud, and that user’s access via a web browser, a dedicated desktop client, or an API that integrates with a desktop or mobile operating system. In most cases, SaaS users pay a monthly or annual subscription fee; some may offer ‘pay-as-you-go’ pricing based on your actual usage.

In addition to the cost savings, time-to-value, and scalability benefits of cloud, SaaS offers the following:

- **Automatic upgrades:** With SaaS, users take advantage of new features as soon as the provider adds them, without having to orchestrate an on-premises upgrade.
- **Protection from data loss:** Because SaaS stores application data in the cloud with the application, users don't lose data if their device crashes or breaks.

SaaS is the primary delivery model for most commercial software today—there are hundreds of thousands of SaaS solutions available, from the most focused industry and departmental applications to powerful enterprise software database and AI (artificial intelligence) software.

## Cloud Service Providers

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Oracle
- IBM cloud
- Salesforce
- 

## Amazon Web Services

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis. Oftentimes, clients will use this in combination with autoscaling (a process that allows a client to use more computing in times of high application usage, and then scale down to reduce costs when there is less traffic). These cloud computing web services provide various services related to networking, computing, storage, middleware, IoT and other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware, and operating systems.

One of the foundational services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, with extremely high availability, which can be interacted with over the internet via REST APIs, a CLI or the AWS console. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard disk /SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

AWS services are delivered to customers via a network of AWS server farms located throughout the world. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either.

Amazon provides select portions of security for subscribers (e.g., physical security of the data centers) while other aspects of security are the responsibility of the subscriber (e.g., account management, vulnerability scanning, patching). AWS operates for many global geographical regions including seven in North America.

Amazon markets AWS to subscribers as a way of obtaining large-scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. As of 2021 Q4, AWS has 33% market share for cloud infrastructure while the next two competitors Microsoft Azure and Google Cloud have 21%, and 10% respectively, according to Synergy Group.

## Why AWS?

- **Easy to use:**

AWS is designed to allow application providers, ISVs, and vendors to host your applications quickly and securely – whether an existing application or a new SaaS-based application. You can use the AWS Management Console or well-documented web services APIs to access AWS's application hosting platform.

- **Flexible:**

AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. This eases the migration process for existing applications while preserving options for building new solutions.

- **Cost-effective:**

You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments. For more information on comparing the costs of other hosting alternatives with AWS, see the AWS Economics Center.

- **Reliable:**

With AWS, you take advantage of a scalable, reliable, and secure global computing infrastructure, the virtual backbone of Amazon.com's multi-billion-dollar online business that has been honed for over a decade.

- **Scalable and High performance:**

Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

- **Secure:**

Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

## List of AWS Services

Amazon, the preeminent cloud vendor, broke new ground by establishing the first cloud computing service, Amazon EC2, in 2008. AWS offers more solutions and features than any other provider and has free tiers with access to the AWS Console, where users can centrally control their ministrations.

Designed around ease-of-use for various skill sets, AWS is tailored for those unaccustomed to software development utilities. Web applications can be deployed in minutes with AWS facilities, without provisioning servers or writing additional code.

- Amazon EC2 (Elastic Compute Cloud)
- Amazon RDS (Relational Database Services)
- Amazon S3 (Simple Storage Service)
- Amazon Lambda
- Amazon Cognito
- Amazon Glacier
- Amazon SNS (Simple Notification Service)
- Amazon VPC (Virtual Private Cloud)
- Amazon Lightsail
- Amazon CloudWatch
- Amazon Cloud9
- Amazon Elastic Beanstalk
- Amazon CodeCommit
- Amazon IAM (Identity and Access Management)
- Amazon Inspector
- Amazon Kinesis
- Amazon Dynamo DB
- Amazon Codecatalyst
- Amazon Kinesis
- AWS Athena
- AWS Amplify
- AWS Quicksight
- AWS Cloudformation

## **Amazon EC2**

Amazon Elastic Compute Cloud (EC2) is a part of Amazon.com's cloud-computing platform, Amazon Web Services (AWS), that allows users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server-instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy. In November 2010, Amazon switched its own retail website platform to EC2 and AWS.

Amazon announced a limited public beta test of EC2 on August 25, 2006, offering access on a first-come, first-served basis. Amazon added two new instance types (Large and Extra-Large) on October 16, 2007. On May 29, 2008, two more types were added, High-CPU Medium and High-CPU Extra Large. There were twelve types of instances available.

Amazon added three new features on March 27, 2008, static IP addresses, availability zones, and user selectable kernels. On August 20, 2008, Amazon added Elastic Block Store (EBS). This provides persistent storage, a feature that had been lacking since the service was introduced.

### **Instance types:**

Initially, EC2 used Xen virtualization exclusively. However, on November 6, 2017, Amazon announced the new C5 family of instances that were based on a custom architecture around the KVM hypervisor, called Nitro. Each virtual machine, called an "instance", functions as a virtual private server. Amazon sizes instances based on "Elastic Compute Units". The performance of otherwise identical virtual machines may vary. On November 28, 2017, AWS announced a bare-metal instance type offering marking a remarkable departure from exclusively offering virtualized instance types.

As of January 2019, the following instance types were offered:

- General Purpose: A1, T3, T2, M5, M5a, M4, T3a
- Compute Optimized: C5, C5n, C4
- Memory Optimized: R5, R5a, R4, X1e, X1, High Memory, z1d
- Accelerated Computing: P3, P2, G3, F1
- Storage Optimized: H1, I3, D2

As of April 2018, the following payment methods by instance were offered:

- On-demand: pay by the hour without commitment.
- Reserved: rent instances with one-time payment receiving discounts on the hourly charge.
- Spot: bid-based service runs the jobs only if the spot price is below the bid specified by bidder. The spot price is claimed to be supply-demand based, however a 2011 study concluded that the price was generally not set to clear the market but was dominated by undisclosed reserve price.

## **Amazon RDS**

**Amazon Relational Database Service** (or **Amazon RDS**) is a distributed relational database service by Amazon Web Services (AWS). It is a web service running "in the cloud" designed to simplify the setup, operation, and scaling of a relational database for use in applications. Administration processes like patching the database software, backing up databases and enabling point-in-time recovery are managed automatically. Scaling storage and compute resources can be performed by a single API call to the AWS control plane on-demand. AWS does not offer an SSH connection to the underlying virtual machine as part of the managed service.

### **Multiple Availability Zone (AZ) Deployment**

In May 2010 Amazon announced Multi-Availability Zone deployment support. Amazon RDS Multi-Availability Zone (AZ) allows users to automatically provision and maintain a synchronous physical or logical "standby" replica, depending on database engine, in a different Availability Zone (independent infrastructure in a physically separate location). Multi-AZ database instance can be developed at creation time or modified to run as a multi-AZ deployment later. Multi-AZ deployments aim to provide enhanced availability and data durability for MySQL, MariaDB, Oracle, PostgreSQL and SQL Server instances and are targeted for production environments. In the event of planned database maintenance or unplanned service disruption, Amazon RDS automatically fails over to the up-to-date standby, allowing database operations to resume without administrative intervention.

Multi-AZ RDS instances are optional and have a cost associated with them. When creating a RDS instance, the user is asked if they would like to use a multi-AZ RDS instance. In Multi-AZ RDS deployments backups are done in the standby instance so I/O activity is not suspended any time, but users may experience elevated latencies for a few minutes during backups.

### **Read replicas.**

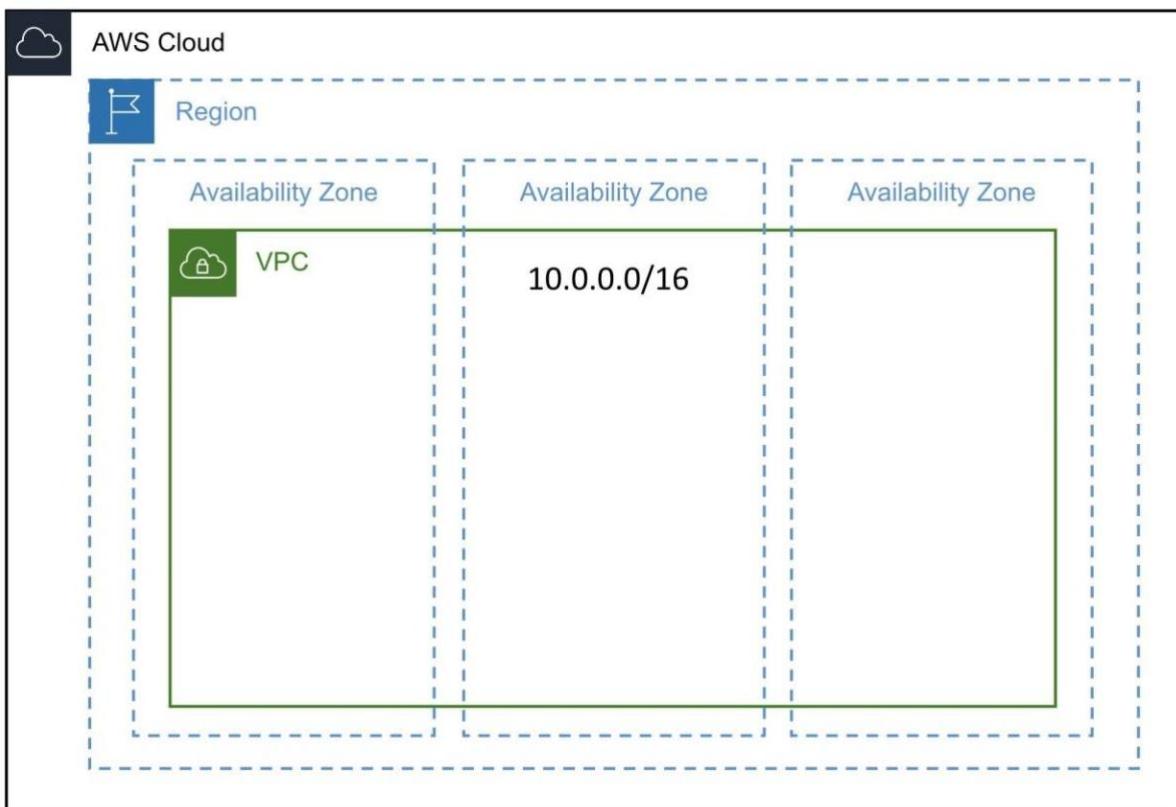
Read replicas allow different use cases such as scale in for read-heavy database workloads. There are up to five replicas available for MySQL, MariaDB, and PostgreSQL. Instances use the native, asynchronous replication functionality of their respective database engines. They have no backups configured by default and are accessible and can be used for read scaling. MySQL and MariaDB read replicas and can be made writeable again since October 2012; PostgreSQL read replicas do not support it. Replicas are done at database instance level and do not support replication at database or table level.

### **Performance metrics and monitoring**

Performance metrics for Amazon RDS are available from the AWS Management Console or the Amazon CloudWatch API. In December 2015, Amazon announced an optional enhanced monitoring feature that provides an expanded set of metrics for the MySQL, MariaDB, and Aurora database engines.

## Amazon VPC

Amazon Virtual Private Cloud (VPC) is a commercial cloud computing service that provides a virtual private cloud, by provisioning a logically isolated section of Amazon Web Services (AWS) Cloud. Enterprise customers are able to access the Amazon Elastic Compute Cloud (EC2) over an IPsec based virtual private network. Unlike traditional EC2 instances which are allocated internal and external IP numbers by Amazon, the customer can assign IP numbers of their choosing from one or more subnets.



Amazon Web Services launched Amazon Virtual Private Cloud on 26 August 2009, which allows the Amazon Elastic Compute Cloud service to be connected to legacy infrastructure over an IPsec VPN. In AWS, the basic VPC is free to use, with users being charged by usage for additional features. EC2 and RDS instances running in a VPC can also be purchased using Reserved Instances, however will have a limitation on resources being guaranteed. [citation needed]

IBM Cloud launched IBM Cloud VPC on 4 June 2019, provides an ability to manage virtual machine-based compute, storage, and networking resources. Pricing for IBM Cloud Virtual Private Cloud is applied separately for internet data transfer, virtual server instances, and block storage used within IBM Cloud VPC.

Google Cloud Platform resources can be provisioned, connected, and isolated in a virtual private cloud (VPC) across all GCP regions. With GCP, VPCs are global resources and subnets within that VPC are regional resources. This allows users to connect zones and regions without the use of additional networking complexity as all data travels, encrypted in transit and at rest, on Google's

own global, private network. Identity management policies and security rules allow for private access to Google's storage, big data, and analytics managed services. VPCs on Google Cloud Platform leverage the security of Google's data centers.

## Amazon S3

Amazon S3 manages data with an object storage architecture which aims to provide scalability, high availability, and low latency with high durability. The basic storage units of Amazon S3 are objects which are organized into buckets. Each object is identified by a unique, user-assigned key. Buckets can be managed using the console provided by Amazon S3, programmatically with the AWS SDK, or the REST application programming interface.

Objects can be up to five terabytes in size. Requests are authorized using an access control list associated with each object bucket and support versioning which is disabled by default. Since buckets are typically the size of an entire file system mount in other systems, this access control scheme is very coarse-grained. In other words, unique access controls cannot be associated with individual files. [citation needed] Amazon S3 can be used to replace static web-hosting infrastructure with HTTP client-accessible objects, index document support and error document support.

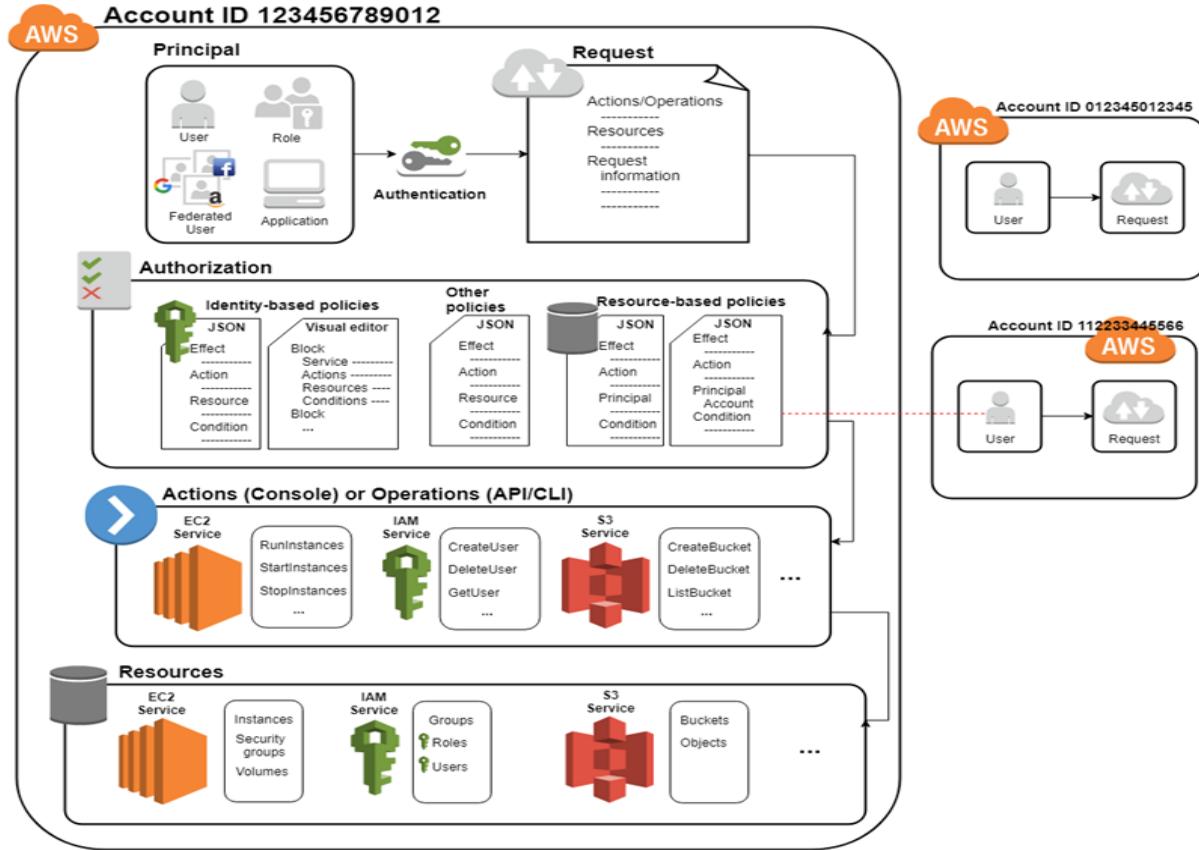
The Amazon AWS authentication mechanism allows the creation of authenticated URLs, valid for a specified amount of time. Every item in a bucket can also be served as a BitTorrent feed. The Amazon S3 store can act as a seed host for a torrent and any BitTorrent client can retrieve the file. This can drastically reduce the bandwidth cost for the download of popular objects. A bucket can be configured to save HTTP log information to a sibling bucket; this can be used in data mining operations.

There are various User Mode File System (FUSE)-based file systems for Unix-like operating systems (for example, Linux) that can be used to mount an S3 bucket as a file system. The semantics of the Amazon S3 file system is not that of a POSIX file system, so the file system may not behave entirely as expected.



## Amazon IAM

IAM provides the infrastructure necessary to control authentication and authorization for your AWS account. The IAM infrastructure is illustrated by the following diagram.



First, a human user or an application uses their sign-in credentials to authenticate with AWS. Authentication is provided by matching the sign-in credentials to a principal (an IAM user, federated user, IAM role, or application) trusted by the AWS account.

Next, a request is made to grant the principal access to resources. Access is granted in response to an authorization request. For example, when you first sign into the console and are on the console home page, you are not accessing a specific service. When you select a service, the request for authorization is sent to that service and it looks to see if your identity is on the list of authorized users, what policies are being enforced to control the level of access granted, and any other policies that might be in effect. Authorization requests can be made by principals within your AWS account or from another AWS account that you trust.

Once authorized, the principal can take action or perform operations on resources in your AWS account. For example, the principal could launch a new Amazon Elastic Compute Cloud instance, modify IAM group membership, or delete Amazon Simple Storage Service buckets.

The previous illustration we used specific terminology to describe how to obtain access to resources. These IAM terms are commonly used when working with AWS

## IAM Resources

The user, group, role, policy, and identity provider objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.

## IAM Identities

The IAM resource objects that are used to identify and group. You can attach a policy to an IAM identity. These include users, groups, and roles.

## IAM Entities

The IAM resource objects that AWS uses for authentication. These include IAM users and roles.

## Principals

A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS. Principals include federated users and assumed roles.

## Human users

Also known as human identities; the people, administrators, developers, operators, and consumers of your applications.

## Workload

A collection of resources and code that delivers business value, such as an application or backend process. Can include applications, operational tools, and components

## AWS Lambda

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, all you need to do is supply your code in one of the language runtimes that Lambda supports.

You organize your code into Lambda functions. The Lambda service runs your function only when needed and scales automatically. You only pay for the compute time that you consume—there is no charge when your code is not running.

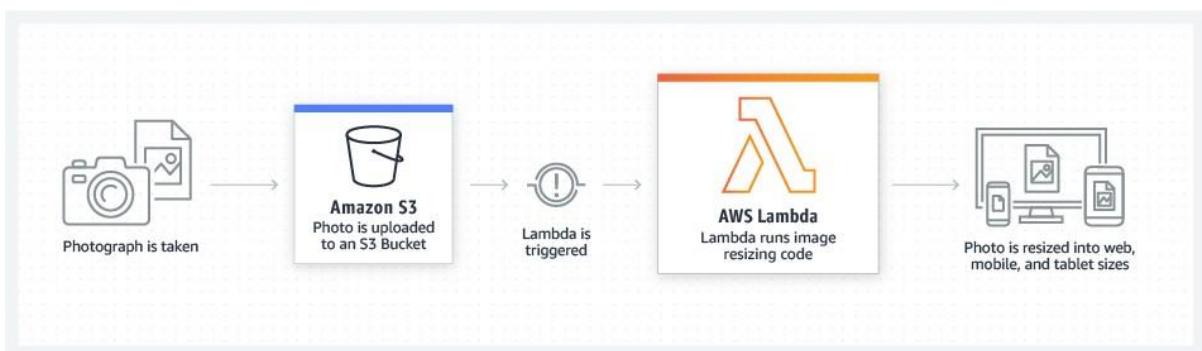
When using Lambda, you are responsible only for your code. Lambda manages the compute fleet that offers a balance of memory, CPU, network, and other resources to run your code. Because Lambda manages these resources, you cannot log in to compute instances or customize the operating system on provided runtimes.

Lambda performs operational and administrative activities on your behalf, including managing capacity, monitoring, and logging your Lambda functions.

If you do need to manage your compute resources, AWS has other compute services to consider, such as:

- AWS App Runner builds and deploys containerized web applications automatically, load balances traffic with encryption, scales to meet your traffic needs, and allows for the configuration of how services are accessed and communicate with other AWS applications in a private Amazon VPC.
- AWS Fargate with Amazon ECS runs containers without having to provision, configure, or scale clusters of virtual machines.
- Amazon EC2 lets you customize operating system, network and security settings, and the entire software stack. You are responsible for provisioning capacity, monitoring fleet health and performance, and using Availability Zones for fault tolerance.

You can use environment variables to adjust your function's behavior without updating code. An environment variable is a pair of strings that is stored in a function's version-specific configuration. The Lambda runtime makes environment variables available to your code and sets additional environment variables that contain information about the function and invocation request.

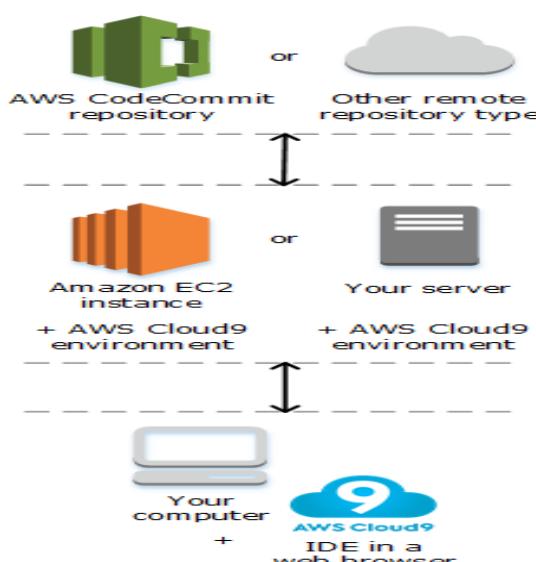


## AWS Cloud9

AWS Cloud9 is an integrated development environment, or *IDE*.

The AWS Cloud9 IDE offers a rich code-editing experience with support for several programming languages and runtime debuggers, and a built-in terminal. It contains a collection of tools that you use to code, build, run, test, and debug software, and helps you release software to the cloud.

You access the AWS Cloud9 IDE through a web browser. You can configure the IDE to your preferences. You can switch color themes, bind shortcut keys, enable programming language-specific syntax coloring and code formatting, and more.



## **Environments and computing resources**

Behind the scenes, there are a couple of ways you can connect your environments to computing resources:

- You can instruct AWS Cloud9 to create an Amazon EC2 instance, and then connect the environment to that newly created EC2 instance. This type of setup is called an *EC2 environment*.
- You can instruct AWS Cloud9 to connect an environment to an existing cloud compute instance or to your own server. This type of setup is called an *SSH environment*.

EC2 environments and SSH environments have some similarities and some differences. If you're new to AWS Cloud9, we recommend that you use an EC2 environment because AWS Cloud9 takes care of much of the configuration for you. As you learn more about AWS Cloud9, and want to understand these similarities and differences better, see EC2 environments compared with SSH environments in AWS Cloud9.

## **AWS Elastic BeanStalk**

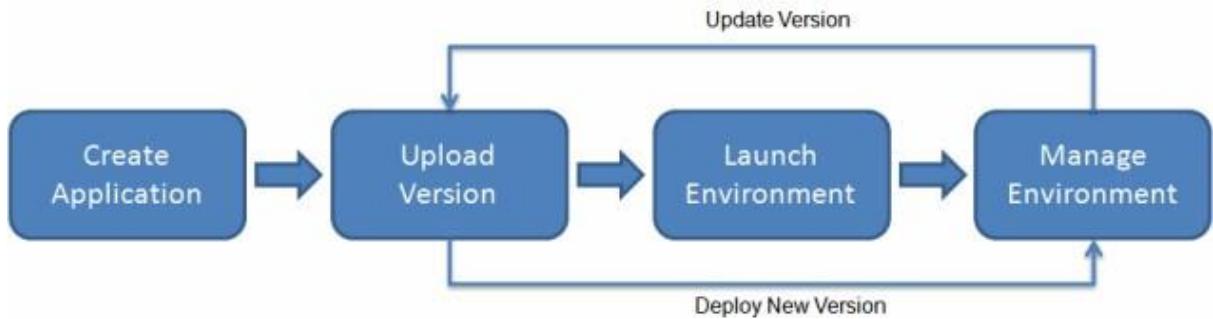
Amazon Web Services (AWS) comprises over one hundred services, each of which exposes an area of functionality. While the variety of services offers flexibility for how you want to manage your AWS infrastructure, it can be challenging to figure out which services to use and how to provision them.

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application. You can interact with Elastic Beanstalk by using the Elastic Beanstalk console, the AWS Command Line Interface (AWS CLI), or `eb`, a high-level CLI designed specifically for Elastic Beanstalk.

To learn more about how to deploy a sample web application using Elastic Beanstalk, see [Getting Started with AWS: Deploying a Web App](#). You can also perform most deployment tasks, such as changing the size of your fleet of Amazon EC2 instances or monitoring your application, directly from the Elastic Beanstalk web interface (console).

To use Elastic Beanstalk, you create an application, upload an application version in the form of an application source bundle (for example, a Java .war file) to Elastic Beanstalk, and then provide some information about the application. Elastic Beanstalk automatically launches an environment and creates and configures the AWS resources needed to run your code. After your environment is launched, you can then manage your environment and deploy new application versions. The following diagram illustrates the workflow of Elastic Beanstalk.

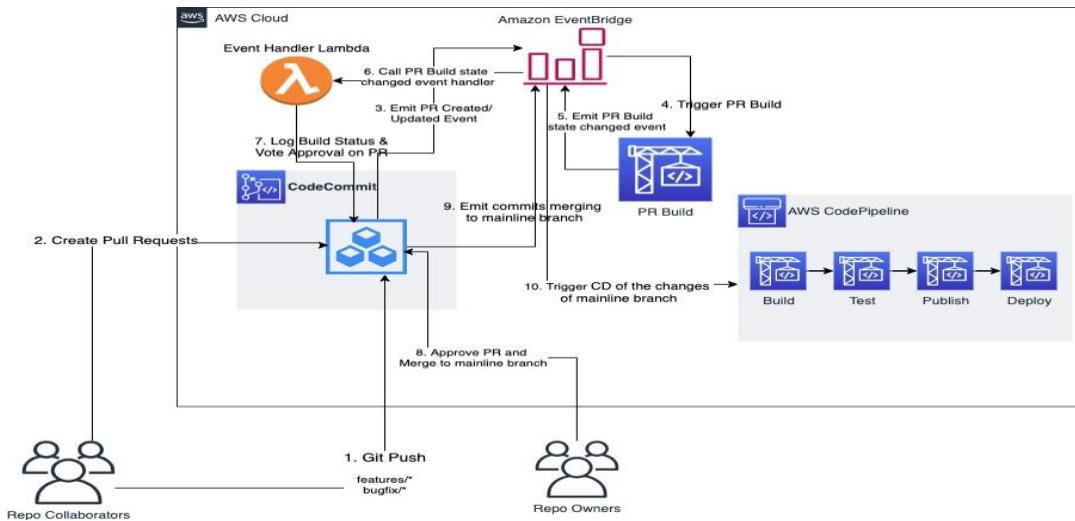


## AWS CodeCommit

CodeCommit is a secure, highly scalable, managed source control service that hosts private Git repositories. CodeCommit eliminates the need for you to manage your own source control system or worry about scaling its infrastructure. You can use CodeCommit to store anything from code to binaries. It supports the standard functionality of Git, so it works seamlessly with your existing Git-based tools.

With CodeCommit, you can:

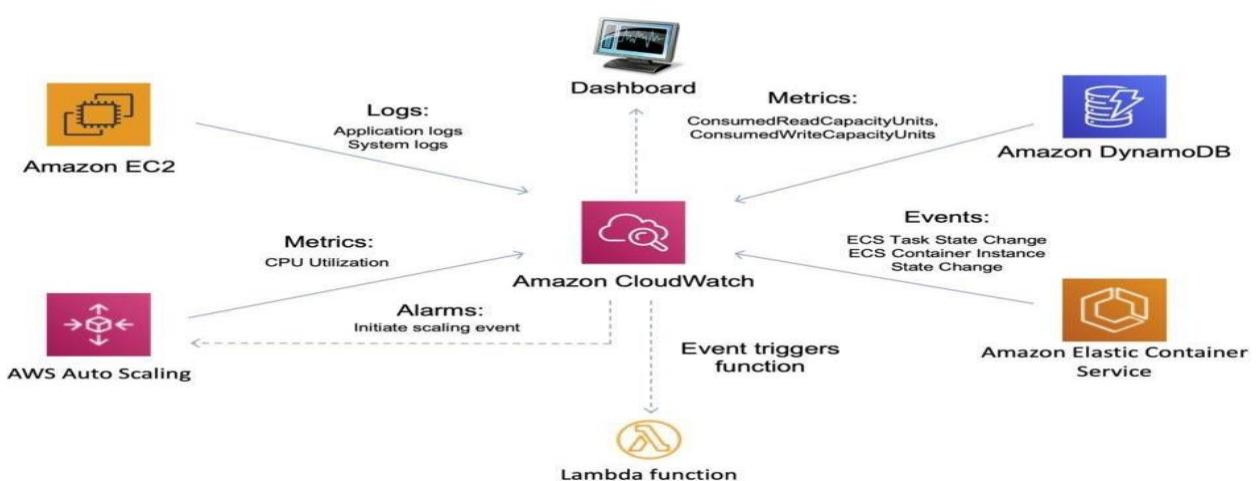
- **Benefit from a fully managed service hosted by AWS.** CodeCommit provides high service availability and durability and eliminates the administrative overhead of managing your own hardware and software. There is no hardware to provision and scale and no server software to install, configure, and update.
- **Store your code securely.** CodeCommit repositories are encrypted at rest as well as in transit.
- **Work collaboratively on code.** CodeCommit repositories support pull requests, where users can review and comment on each other's code changes before merging them to branches; notifications that automatically send emails to users about pull requests and comments; and more.
- **Easily scale your version control projects.** CodeCommit repositories can scale up to meet your development needs. The service can handle repositories with large numbers of files or branches, large file sizes, and lengthy revision histories.
- **Store anything, anytime.** CodeCommit has no limit on the size of your repositories or on the file types you can store.
- **Integrate with other AWS and third-party services.** CodeCommit keeps your repositories close to your other production resources in the AWS Cloud, which helps increase the speed and frequency of your development lifecycle. It is integrated with IAM and can be used with other AWS services and in parallel with other repositories. For more information, see Product and service integrations with AWS CodeCommit.
- **Easily migrate files from other remote repositories.** You can migrate to CodeCommit from any Git-based repository.
- **Use the Git tools you already know.** CodeCommit supports Git commands as well as its own AWS CLI commands and APIs.



## Amazon CloudWatch

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications. The CloudWatch home page automatically displays metrics about every AWS service you use. You can additionally create custom dashboards to display metrics about your custom applications and display custom collections of metrics that you choose.

You can create alarms that watch metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached. For example, you can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances and then use that data to determine whether you should launch additional instances to handle increased load. You can also use this data to stop underused instances to save money. With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.



## **Amazon EBS (Elastic Block Store)**

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices. You can mount these volumes as devices on your instances. EBS volumes that are attached to an instance are exposed as storage volumes that persist independently from the life of the instance. You can create a file system on top of these volumes or use them in any way you would use a block device (such as a hard drive). You can dynamically change the configuration of a volume attached to an instance.

We recommend Amazon EBS for data that must be quickly accessible and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long, continuous reads and writes.

With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the Projecting Costs Section of the Amazon Elastic Block Store page.

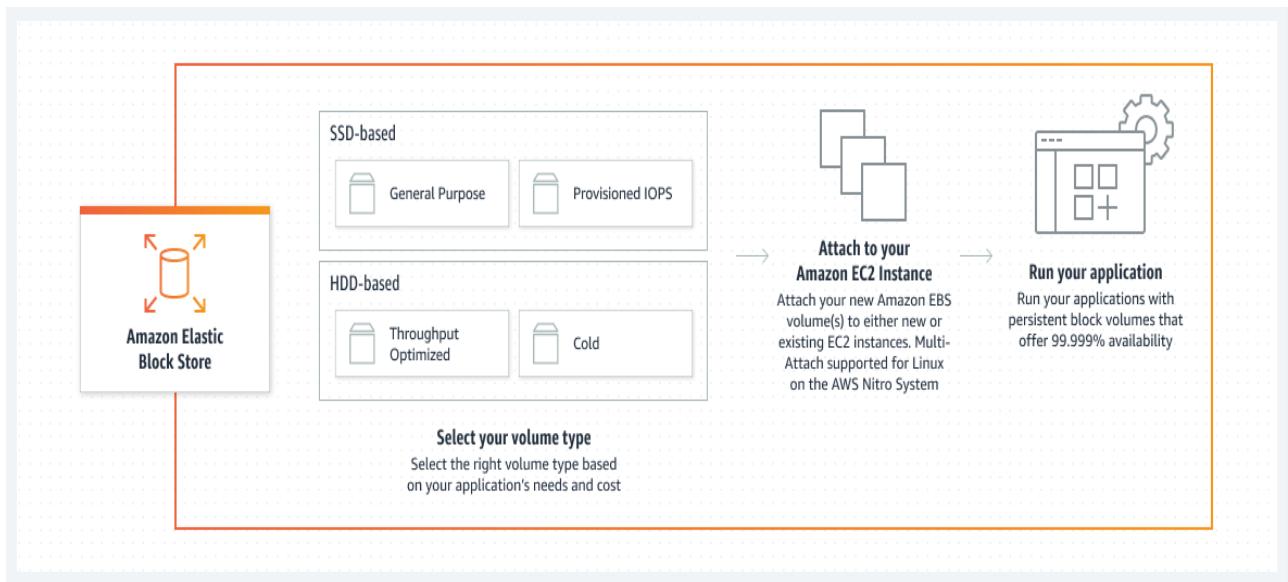
### **Features of Amazon EBS**

- You create an EBS volume in a specific Availability Zone, and then attach it to an instance in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that Region. You can copy snapshots to other Regions and then restore them to new volumes there, making it easier to leverage multiple AWS Regions for geographical expansion, data center migration, and disaster recovery.
- Amazon EBS provides the following volume types: General Purpose SSD, Provisioned IOPS SSD, Throughput Optimized HDD, and Cold HDD. For more information, see EBS volume types.

The following is a summary of performance and use cases for each volume type.

- General Purpose SSD volumes (gp2 and gp3) balance price and performance for a wide variety of transactional workloads. These volumes are ideal for use cases such as boot volumes, medium-size single instance databases, and development and test environments.
- Provisioned IOPS SSD volumes (io1 and io2) are designed to meet the needs of I/O-intensive workloads that are sensitive to storage performance and consistency. They provide a consistent IOPS rate that you specify when you create the volume. This enables you to predictably scale to tens of thousands of IOPS per instance. Additionally, io2 volumes provide the highest levels of volume durability.
- Throughput Optimized HDD volumes (st1) provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing.

- Cold HDD volumes (sc1) provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential, cold-data workloads. If you require infrequent access to your data and are looking to save costs, these volumes provide inexpensive block storage.
- You can create your EBS volumes as encrypted volumes, in order to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. Encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see Amazon EBS encryption.
- Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need.



**Fig.** High-Performance Block Storage

## Amazon Aurora

Amazon Aurora (Aurora) is a fully managed relational database engine that's compatible with MySQL and PostgreSQL. You already know how MySQL and PostgreSQL combine the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases. The code, tools, and applications you use today with your existing MySQL and PostgreSQL databases can be used with Aurora. With some workloads, Aurora can deliver up to five times the throughput of MySQL and up to three times the throughput of PostgreSQL without requiring changes to most of your existing applications.

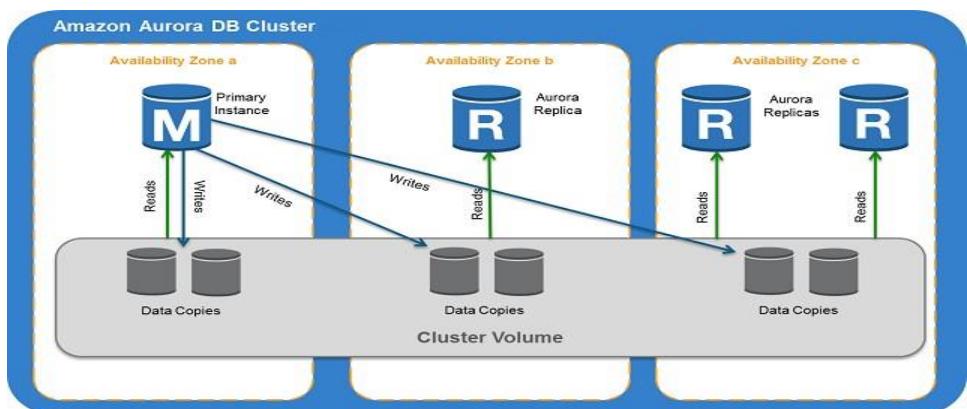
Aurora includes a high-performance storage subsystem. Its MySQL- and PostgreSQL-compatible database engines are customized to take advantage of that fast distributed storage. The underlying storage grows automatically as needed. An Aurora cluster volume can grow to a maximum size of 128 tebibytes (TiB). Aurora also automates and standardizes database clustering and replication, which are typically among the most challenging aspects of database configuration and administration.

Aurora is part of the managed database service Amazon Relational Database Service (Amazon RDS). Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. If you are not already familiar with Amazon RDS, see the *Amazon Relational Database Service User Guide*.

The following points illustrate how Amazon Aurora relates to the standard MySQL and PostgreSQL engines available in Amazon RDS:

- You choose Aurora MySQL or Aurora PostgreSQL as the DB engine option when setting up new database servers through Amazon RDS.
- Aurora takes advantage of the familiar Amazon Relational Database Service (Amazon RDS) features for management and administration. Aurora uses the Amazon RDS AWS Management Console interface, AWS CLI commands, and API operations to handle routine database tasks such as provisioning, patching, backup, recovery, failure detection, and repair.
- Aurora management operations typically involve entire clusters of database servers that are synchronized through replication, instead of individual database instances. The automatic clustering, replication, and storage allocation make it simple and cost-effective to set up, operate, and scale your largest MySQL and PostgreSQL deployments.

You can bring data from Amazon RDS for MySQL and Amazon RDS for PostgreSQL into Aurora by creating and restoring snapshots, or by setting up one-way replication. You can use push-button migration tools to convert your existing RDS for MySQL and RDS for PostgreSQL applications to Aurora.

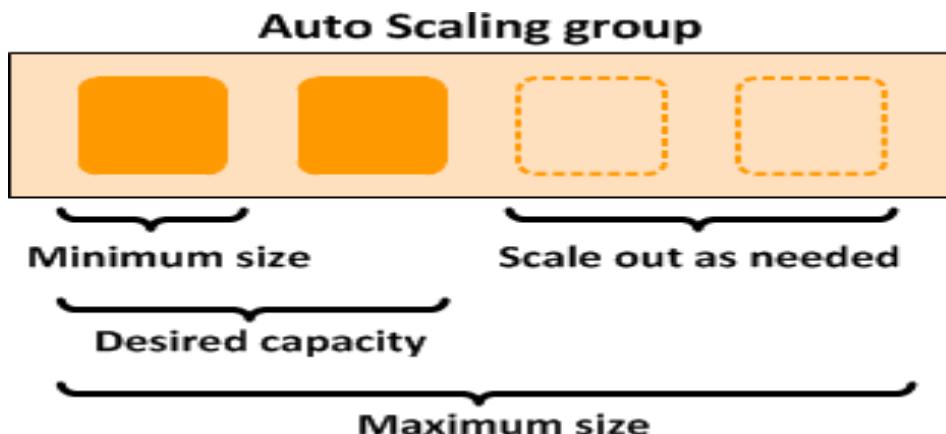


**Fig.**Amazon Auora DB Clusters

## AWS Autoscaling

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, the following Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.



### Auto scaling benefits

Adding Amazon EC2 Auto Scaling to your application architecture is one way to maximize the benefits of the AWS Cloud. When you use Amazon EC2 Auto Scaling, your applications gain the following benefits:

- Better fault tolerance. Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Amazon EC2 Auto Scaling can launch instances in another one to compensate.
- Better availability. Amazon EC2 Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.
- Better cost management. Amazon EC2 Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are needed and terminating them when they aren't.

# IMPLEMENTATION

## Steps to perform

### Step-1: VPC

1(a): Go to VPC and create VPC by giving its name and ipv4 CIDR and click on create vpc.

The screenshot shows the 'Create VPC' configuration page. Under 'VPC settings', the 'Resources to create' section is set to 'VPC only'. The 'Name tag - optional' field contains 'my-vpc-01'. The 'IPv4 CIDR block' section shows 'IPv4 CIDR manual input' selected, with '10.0.0.0/24' entered. At the bottom, there are tabs for 'IPv6 CIDR block', 'Tenancy', and 'Tags', followed by a 'Create VPC' button.

The screenshot shows the 'Create VPC' configuration page. Under 'VPC settings', the 'IPv6 CIDR block' section is set to 'No IPv6 CIDR block'. The 'Tenancy' section is set to 'Default'. The 'Tags' section contains a tag 'vpcproject' with value 'projectvpc'. At the bottom, there are tabs for 'IPv6 CIDR block', 'Tenancy', and 'Tags', followed by a 'Create VPC' button.

**vpc-05d76391d5ead4cbf**

Details	Info
VPC ID vpc-05d76391d5ead4cbf	State Available
Tenancy Default	DHCP option set dopt-0a0ee4b88a5dee7de
Default VPC No	IPv4 CIDR 10.0.0.0/24
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -
	DNS hostnames Disabled
	Main route table rtb-0ae13eed7b948e783
	IPv6 pool -
	DNS resolution Enabled
	Main network ACL acl-0fff4e9a38618c633
	IPv6 CIDR (Network border group) -
	Owner ID 539979324382

### 1(b): Edit route table name

Name	Route table ID	Explicit subnet associations	Edge associations	Main
projectrt	rtb-0bf3726ea0778a6ac	-	-	Yes
projectrt	rtb-0ae13eed7b948e783	-	-	Yes

### 1(c): Creating a subnet using the created vpc with CIDR:10.0.0.0/24:

**Create subnet**

**VPC**

VPC ID  
Create subnets in this VPC.  
vpc-05d76391d5ead4cbf (projectvpc)

**Associated VPC CIDRs**

IPv4 CIDRs  
10.0.0.0/24

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

Subnet name  
Create a tag with a key of 'Name' and a value that you specify.

Subnet 1 of 1

Subnet name  
Create a tag with a key of 'Name' and a value that you specify.

Subnet

The name can be up to 256 characters long.

Availability Zone Info  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Mumbai) / ap-south-1a

IPv4 CIDR block Info  
10.0.0.0/28

Tags - optional

Key	Value - optional
Subnet	Subnet
Name	Subnet

Add new tag  
You can add 48 more tags.

Remove

CloudShell Feedback Language © 2025, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

You have successfully created 1 subnet: subnet-05ed097533300ac9f

Subnets (1) Info

Name	Subnet ID	State	VPC	IPv4 CIDR
Subnet	subnet-05ed097533300ac9f	Available	vpc-05d76391d5ead4cbf   pro...	10.0.0.0/28

Select a subnet

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

## 1(d): Edit subnet association in route tables.

VPC > Route tables > rtb-0ae13eed7b948e783 > Edit subnet associations

### Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Subnet	subnet-05ed097533300ac9f	10.0.0.0/28	-	Main (rtb-0ae13eed7b948e783 / pro...)

Selected subnets

subnet-05ed097533300ac9f / Subnet X

Cancel Save associations

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

## 1(e) Create internet gateway

The screenshot shows the 'Create internet gateway' page in the AWS VPC service. At the top, there's a navigation bar with 'Services' and a search bar. Below it, a breadcrumb trail shows 'VPC > Internet gateways > Create internet gateway'. The main form has a section titled 'Internet gateway settings' where a 'Name tag' is specified as 'projectigw'. A 'Tags - optional' section contains one tag: 'Name' with value 'projectigw'. At the bottom right of the form is a large orange 'Create internet gateway' button.

This screenshot is similar to the previous one but includes an additional 'Add new tag' button at the bottom of the 'Tags - optional' section, which allows for adding more tags.

The screenshot shows the 'Internet gateways' page in the AWS VPC service. A green success message at the top states: 'The following internet gateway was created: igw-09bc0104b9f4330c6 - projectigw. You can now attach to a VPC to enable the VPC to communicate with the internet.' Below this, the 'igw-09bc0104b9f4330c6 / projectigw' gateway is listed. The 'Details' section shows its ID as 'igw-09bc0104b9f4330c6', state as 'Detached', and owner as '539979324382'. The 'Tags' section lists a single tag 'Name: projectigw'. On the left sidebar, under 'Your VPCs', the 'Internet gateways' section is expanded, showing the newly created gateway.

## 1(f): Edit routes

The screenshot shows the 'Edit routes' page for a specific route table. It lists two routes:

- Destination: 10.0.0.0/24, Target: local, Status: Active, Propagated: No.
- Destination: 0.0.0.0/0, Target: igw-09bc0104b9f4330c6, Status: -, Propagated: No. A 'Remove' button is next to it.

Buttons at the bottom include 'Add route', 'Cancel', 'Preview', and 'Save changes'.

## Step-2: EC2 Instance

### 2(a) Launch instance

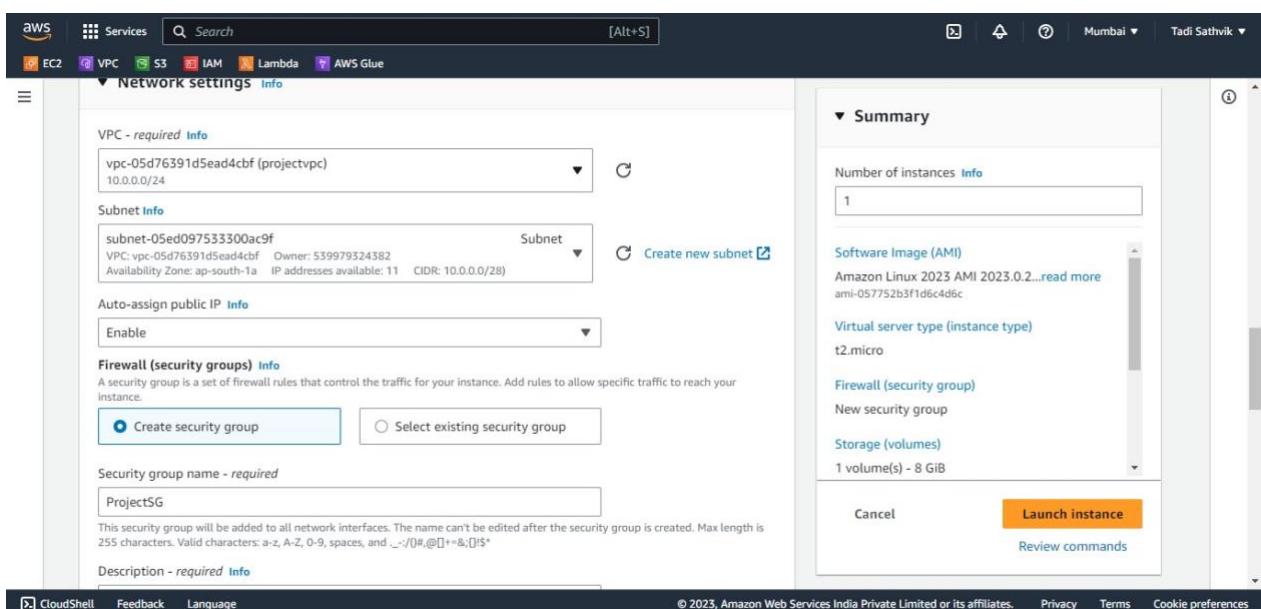
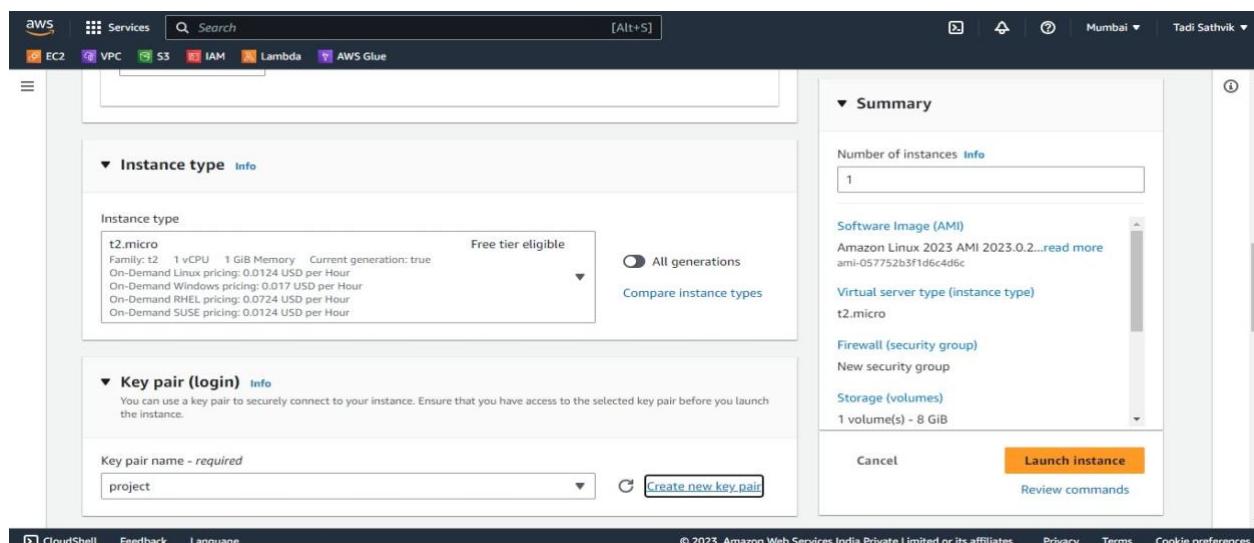
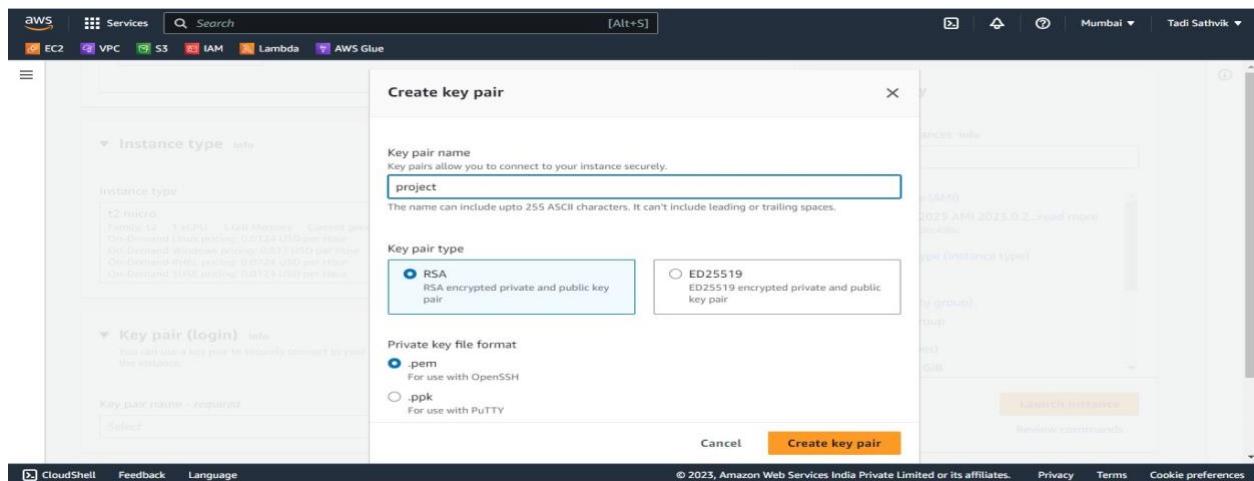
The screenshot shows the 'Launch an instance' page. Configuration fields include:

- Name and tags:** Name is set to 'Projectinstance'.
- Application and OS Images (Amazon Machine Image):** An AMI is selected: 'Amazon Linux 2 Kernel 5.10 AMI...'. Other options like 'Windows' and 'Red Hat' are also shown.
- Virtual server type (instance type):** 't2.micro' is selected.
- Storage (volumes):** 1 volume(s) - 8 GiB.

Buttons at the bottom include 'Cancel', 'Launch instance', and 'Review commands'.

The screenshot shows the 'Quick Start' section of the EC2 interface, specifically the 'Amazon Machine Image (AMI)' selection area. It displays:

- A search bar: 'Search our full catalog including 1000s of application and OS images'.
- A 'Quick Start' heading.
- A grid of AMI icons: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and a 'Browse more AMIs' option.
- Detailed information for the selected 'Amazon Linux 2023 AMI': AMI ID, Virtualization, ENA enabled, Root device type, and a note about Free tier eligibility.
- Additional details: Description (Amazon Linux 2023 AMI 2023.0.20230614.0 x86\_64 HVM kernel-6.1), Architecture (64-bit (x86)), and AMI ID (ami-057752b3f1d6c4d6c).
- A summary panel on the right with identical configuration settings: Number of instances (1), Software Image (Amazon Linux 2023 AMI 2023.0.2...), Virtual server type (t2.micro), Firewall (New security group), Storage (1 volume(s) - 8 GiB), and 'Launch instance' and 'Review commands' buttons.



The screenshot shows the AWS EC2 Security Groups configuration. It displays two security group rules:

- Security group rule 1 (TCP, 22, 0.0.0.0/0):** Type ssh, Protocol TCP, Port range 22, Source Anywhere, Description e.g. SSH for admin desktop.
- Security group rule 2 (All, All, Multiple sources):** Type All traffic, Protocol All, Port range All, Source Anywhere, Description e.g. SSH for admin desktop.

At the bottom left is a "Add security group rule" button. On the right, the "Summary" section shows 1 instance, Amazon Linux 2023 AMI 2023.0.2, t2.micro instance type, and 1 volume(s) - 8 GiB storage. Buttons for "Launch instance" and "Review commands" are present.

The screenshot shows the AWS EC2 Launch instance configuration. It includes sections for "Advanced network configuration" and "Configure storage".

**Configure storage:** 1x 8 GiB gp2 Root volume (Not encrypted). A note indicates free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage.

**Advanced details:** 0 x File systems.

The "Summary" section on the right shows 1 instance, Amazon Linux 2023 AMI 2023.0.2, t2.micro instance type, and 1 volume(s) - 8 GiB storage. Buttons for "Launch instance" and "Review commands" are present.

The screenshot shows the AWS EC2 Instances page. A sidebar on the left lists navigation options like EC2 Dashboard, EC2 Global View, Events, Limits, Instances, Images, and AMIs. The main area shows a table for Instances (1) with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Projectinstance	i-0e7cb11050aa3a55d	Pending	t2.micro	-	No alarms	ap-south-1a

A modal window titled "Select an instance" is open at the bottom, showing a single entry: "Projectinstance".

## 2(b): Connect to the instance

The screenshot shows the 'Connect to instance' page for instance i-0e7cb11050aa3a55d. It displays two main connection methods: 'EC2 Instance Connect' (selected) and 'EC2 Instance Connect Endpoint'. Below these are fields for 'Public IP address' (3.109.183.149), 'User name' (ec2-user), and a note about the AMI user name. At the bottom, there are links for CloudShell, Feedback, Language, and a copyright notice.

The screenshot shows the 'Connect to instance' page for instance i-0e7cb11050aa3a55d. It provides step-by-step instructions for using an SSH client: 1. Open an SSH client, 2. Locate your private key file (project.pem), 3. Run chmod 400 project.pem, 4. Connect to the instance using its Public IP (3.109.183.149). It also includes an example command (ssh -i "project.pem" ec2-user@3.109.183.149) and a note about the user name. The bottom section is identical to the previous screenshot.

## 2(c): Modify IAM role

The screenshot shows the 'Modify IAM role' page for instance i-0e7cb11050aa3a55d. It allows attaching a new IAM role. A warning message states that if no role is chosen, the current role will be removed. At the bottom, there are 'Cancel' and 'Update IAM role' buttons.

**Select trusted entity**

**Trusted entity type**

- AWS service
- AWS account
- Web identity
- SAML 2.0 federation
- Custom trust policy

**Use case**

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Common use cases**

- EC2

**Add permissions**

**Permissions policies (Selected 1/861)**

Choose one or more policies to attach to your new role.

Policy name	Type	Description
AWSGlueServiceRole...	Custom...	This policy will be used for Glue Crawler and Job execution. Please do N...
AWSLambdaBasicExe...	Custom...	
AdministratorAccess	AWS m...	Provides full access to AWS services and resources.

**Name, review, and create**

**Role details**

**Role name**

Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+-, @-' characters.

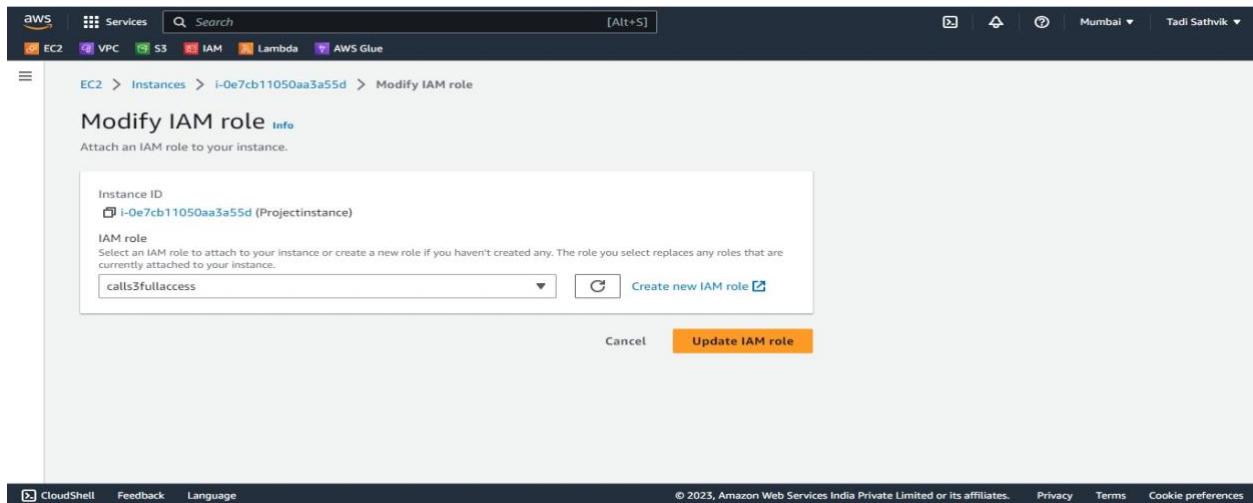
**Description**

Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and '+-, @-' characters.

**Step 1: Select trusted entities**

**Edit**



2(d): Now open window power shell and do below commands for the image.

```

ps -o user=ec2-user,pid=10-0-0-11|grep https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-10-0-0-11 ~]$ curl -v https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-10-0-0-11 ~]$ sudo yum update
Last metadata expiration check: 0:05:00 ago on Mon Jun 26 09:13:03 2023.
Dependencies resolved.
Nothing to do.
[ec2-user@ip-10-0-0-11 ~]$ sudo su
[root@ip-10-0-0-11 ec2-user]# yum install httpd -y
Last metadata expiration check: 0:06:06 ago on Mon Jun 26 09:13:03 2023.
Dependencies resolved:
----- Package Architecture Version
Repository Size
httpd x86_64 2.4.56-1.amzn2023 amazonlinux 48 k
Installing:
apr x86_64 1.7.2-2.amzn2023_0.2 amazonlinux 129 k
apr-util x86_64 1.6.3-1.amzn2023_0.1 amazonlinux 98 k
generic-logos-httdp x86_64 2.4.56-1.amzn2023_0.3 amazonlinux 19 k
httpd-core x86_64 2.4.56-1.amzn2023_0.205 amazonlinux 1.4 M
httpd-fsysteem noarch 2.4.56-1.amzn2023_0.205 amazonlinux 15 k
httpd-tools x86_64 2.4.56-1.amzn2023_0.205 amazonlinux 82 k
libhttpd x86_64 2.4.56-1.amzn2023_0.205 amazonlinux 315 k
mailcap noarch 2.1.49-3.amzn2023_0.3 amazonlinux 33 k
Installing weak dependencies:
apr-util-openssl x86_64 1.6.3-1.amzn2023_0.1 amazonlinux 17 k
mod_httpd x86_64 2.0.11-2.amzn2023_0.3 amazonlinux 158 k
mod_lua x86_64 2.4.56-1.amzn2023_0.203 amazonlinux 62 k
Transaction Summary
----- Install 12 Packages
Total download size: 2.3 M
Installed size: 4.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023_0.1.x86_64.rpm 1.7 MB/s | 98 kB 00:00
(2/12): httpd-core-2.4.56-1.amzn2023.x86_64.rpm 1.6 MB/s | 1.4 MB 00:00
[ec2-user@ip-10-0-0-11 /var/www/html]
----- Total
12 MB/s | 2.3 MB 00:00
unmount transaction check
transaction check succeeded.
umount transaction test
transaction test succeeded.
umount transaction
Preparing :
----- 1/1
Installing : apr-1.7.2-2.amzn2023_0.2.x86_64 1/12
Installing : apr-util-openssl-1.6.3-1.amzn2023_0.1.x86_64 2/12
Installing : apr-util-1.6.3-1.amzn2023_0.1.x86_64 3/12
Installing : mailcap-2.1.49-3.amzn2023_0.3.noarch 4/12
Installing : httpd-tools-2.4.56-1.amzn2023.x86_64 5/12
Installing : generic-logos-httdp-18.0-0-12.amzn2023_0.3.noarch 6/12
Running scriptlet: httpd-fsysteem-2.4.56-1.amzn2023.noarch 7/12
Installing : httpd-fsysteem-2.4.56-1.amzn2023.noarch 7/12
Installing : httpd-core-2.4.56-1.amzn2023.x86_64 8/12
Installing : mod_httpd-2.0.11-2.amzn2023.x86_64 9/12
Installing : mod_lua-2.4.56-1.amzn2023.x86_64 10/12
Installing : librotoli-1.0.9-4.amzn2023_0.2.x86_64 11/12
Installing : httpd-2.4.56-1.amzn2023.x86_64 12/12
Running scriptlet: httpd-2.4.56-1.amzn2023.x86_64 12/12
Verifying : httpd-2.4.56-1.amzn2023.x86_64 1/12
Verifying : apr-util-1.6.3-1.amzn2023_0.1.x86_64 2/12
Verifying : httpd-core-2.4.56-1.amzn2023.x86_64 3/12
Verifying : apr-1.7.2-2.amzn2023_0.2.x86_64 4/12
Verifying : httpd-tools-2.4.56-1.amzn2023.x86_64 5/12
Verifying : librotoli-1.0.9-4.amzn2023_0.2.x86_64 6/12
Verifying : mod_httpd-2.0.11-2.amzn2023.x86_64 7/12
Verifying : apr-util-openssl-1.6.3-1.amzn2023_0.1.x86_64 8/12
Verifying : mod_lua-2.4.56-1.amzn2023.x86_64 9/12
Verifying : httpd-fsysteem-2.4.56-1.amzn2023.noarch 10/12
Verifying : mailcap-2.1.49-3.amzn2023_0.3.noarch 11/12
Verifying : generic-logos-httdp-18.0-0-12.amzn2023_0.3.noarch 12/12
Installed:
apr-1.7.2-2.amzn2023_0.2.x86_64
apr-util-openssl-1.6.3-1.amzn2023_0.1.x86_64
httpd-2.4.56-1.amzn2023.x86_64
httpd-fsysteem-2.4.56-1.amzn2023.noarch
librotoli-1.0.9-4.amzn2023_0.2.x86_64

```

2(e): Go the cmd.

```
PS C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3086]
(C) Microsoft Corporation. All rights reserved.

C:\Users\DLc>cd Downloads
C:\Users\DLc\Downloads>scp -i ./project.pem -r ./test1.jpg ec2-user@3.109.183.149:/var/www/html
scp: /var/www/html/test1.jpg: Permission denied
C:\Users\DLc\Downloads>scp -i ./project.pem -r ./test1.jpg ec2-user@3.109.183.149:/var/www/html
test1.jpg
      100%  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
```

2(f): After getting connected to the instance

```
i-0e7cb11050aa3a55d (Projectinstance)
PublicIPs: 3.109.183.149 PrivateIPs: 10.0.0.11

AWS CloudShell Feedback Language
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Mon Jun 26 09:26:38 2023 from 13.233.177.3
[ec2-user@ip-10-0-0-11 ~]$ sudo su
[root@ip-10-0-0-11 ec2-user]# cd /var/www/html
[root@ip-10-0-0-11 html]# ls
test1.jpg
[root@ip-10-0-0-11 html]#
```

2(g): Now open window power shell and apply below codes for csv files.

```
PS C:\Users\DLc> ssh -i "project.pem" ec2-user@3.109.183.149
Warning: Identity file project.pem not accessible: No such file or directory.
ec2-user@3.109.183.149: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
PS C:\Users\DLc> cd Downloads
PS C:\Users\DLc\Downloads> ssh -i "project.pem" ec2-user@3.109.183.149
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Mon Jun 26 10:15:10 2023 from 13.233.177.3
[ec2-user@ip-10-0-0-11 ~]$ sudo yum update
Last metadata expiration check: 1:12:58 ago on Mon Jun 26 09:13:03 2023.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-0-11 ~]$ ls
[root@ip-10-0-0-11 ec2-user]# cd /var/www/html
[root@ip-10-0-0-11 html]# ls
test1.jpg  username.csv
[root@ip-10-0-0-11 html]#
```

2(h): Open cmd and do the following commands.

```
Command Prompt (Version 10.0.19043.260)
(C) Microsoft Corporation. All rights reserved.

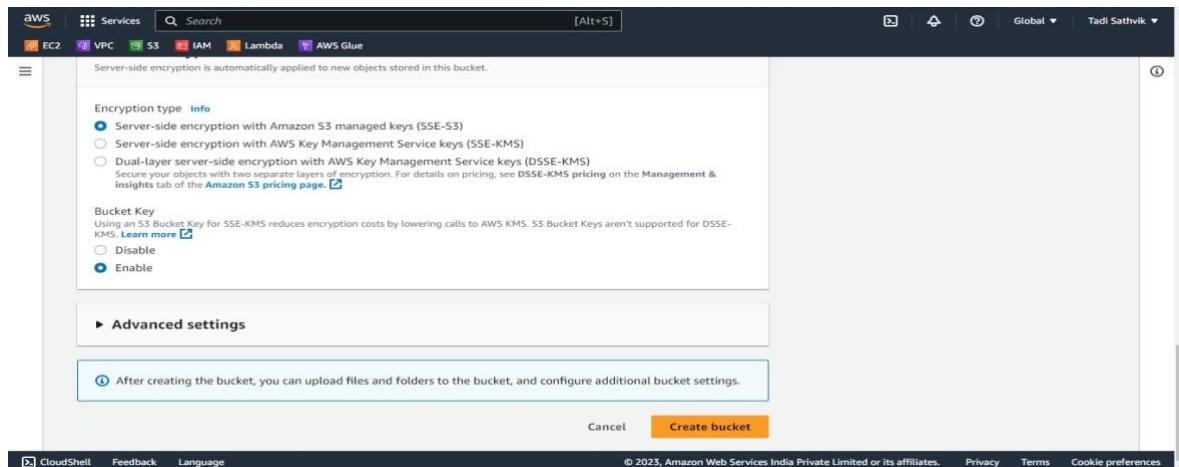
C:\Users\Tadi\Downloads> scp -i ./project.pem -r ./username.csv ec2-user@3.100.103.140:/var/www/html/
      3805 241    7.5KB/s   00:00
C:\Users\Tadi\Downloads>
```

### Step-3: S3

3(a): Creating s3 bucket.

The image consists of three vertically stacked screenshots of the AWS S3 'Create bucket' wizard.

- General configuration:** Shows the 'Bucket name' field set to 'project.aws'. The 'AWS Region' dropdown is set to 'Asia Pacific (Mumbai) ap-south-1'. A note indicates that copy settings from existing buckets are optional.
- Object Ownership:** Shows two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The 'Object Ownership' section shows 'Bucket owner enforced'.
- Block Public Access settings for this bucket:** Shows the 'Block all public access' checkbox checked. A note states that turning this setting on is the same as turning on all four settings below.
- Bucket Versioning:** Shows the 'Bucket Versioning' section with 'Disable' selected.
- Tags (0) - optional:** Shows the note 'You can use bucket tags to track storage costs and organize buckets.' and a 'Add tag' button.
- Default encryption:** Shows the note 'This bucket uses server-side encryption with AWS KMS-managed keys (AES-256). You can also use server-side encryption with your own AWS KMS keys or AWS CloudWatch Metrics Insights logs with AWS Lambda functions to encrypt objects.'



### 3(b): Connection between EC2 and s3 for jpg file.

```

aws [Alt+S] Services Search Global Tadi Sathvik
EC2 VPC S3 IAM Lambda AWS Glue

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Mon Jun 26 09:33:57 2023 from 13.233.177.3
[ec2-user@ip-10-0-0-11 ~]$ sudo su
[root@ip-10-0-0-11 ec2-user]# cd /var/www/html
[root@ip-10-0-0-11 html]# ls
test1.jpg
[root@ip-10-0-0-11 html]# aws s3 ls
2023-06-26 10:02:47 project.aws
[root@ip-10-0-0-11 html]# aws s3 cp test1.jpg s3://project.aws
upload: ./test1.jpg to s3://project.aws/test1.jpg
[root@ip-10-0-0-11 html]#

```

i-0e7cb11050aa3a55d (Projectinstance)  
PublicIPs: 3.109.183.149 PrivateIPs: 10.0.0.11

### 3(c): After successful uploading of image.

Name	Type	Last modified	Size	Storage class
test1.jpg	jpg	June 26, 2023, 15:46:35 (UTC+05:30)	83.1 KB	Standard

### 3(d): Connect EC2 instance to s3 for csv file.

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Mon Jun 26 10:25:55 2023 from 175.101.94.8
[ec2-user@ip-10-0-0-11 ~]$ sudo su
[root@ip-10-0-0-11 ec2-user]# cd /var/www/html
[root@ip-10-0-0-11 html]# ls
test1.jpg username.csv
[root@ip-10-0-0-11 html]# aws s3 ls
2023-06-26 10:02:47 project.awa
[root@ip-10-0-0-11 html]# aws s3 cp username.csv s3://project.awa
[root@ip-10-0-0-11 html]# ls
test1.jpg username.csv
[root@ip-10-0-0-11 html]#

```

i-0e7cb11050aa3a55d (Projectinstance)  
PublicIPs: 3.109.183.149 PrivateIPs: 10.0.0.11

3(e): After successful uploading of csv file.

Amazon S3 > Buckets > project.awa > username.csv

**username.csv** Info

**Properties** Permissions Versions

**Object overview**

Owner	S3 URI
9c36942a31a4c9262450caa1e2d76784acef79c9c5645f38b933bdf0a9861d1c	s3://project.aws/username.csv
AWS Region	Amazon Resource Name (ARN)
Asia Pacific (Mumbai) ap-south-1	arn:aws:s3:::project.aws/username.csv
Last modified	Entity tag (Etag)
June 26, 2023, 16:03:50 (UTC+05:30)	955a7e66ea941a9ee229cf1173ac4442
Size	Object URL
241.0 B	<a href="https://s3.ap-south-1.amazonaws.com/project.aws/username.csv">https://s3.ap-south-1.amazonaws.com/project.aws/username.csv</a>

## Step-4: Lambda

4(a): Creating lambda function

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

**Change default execution role**

Execution role  
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console [IAM](#).

Create a new role with basic Lambda permissions

Use an existing role

Create a new role from AWS policy templates

**Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.**

Lambda will create an execution role named TextExtraction-role-fytuu9s4, with permission to upload logs to Amazon CloudWatch Logs.

**Advanced settings**

**Create function**

**Create function** Info

AWS Serverless Application Repository applications have moved to [Create application](#).

Author from scratch Start with a simple Hello World example.

Use a blueprint Build a Lambda application from sample code and configuration presets for common use cases.

Container image Select a container image to deploy for your function.

**Basic information**

Function name Enter a name that describes the purpose of your function.  
TextExtraction

Runtime Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
Python 3.9

Architecture Choose the instruction set architecture you want for your function code.

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

#### 4(b): Add permissions.

IAM > Roles > TextExtraction-role-fyuu9s4 > Add permissions

Attach policy to TextExtraction-role-fyuu9s4

▼ Current permissions policies (1)

Policy name	Type	Attached entities
<input checked="" type="checkbox"/> AWSLambdaBasicExe...	Custom...	1

Other permissions policies (Selected 3/860)

Policy name	Type	Description
<input type="checkbox"/> AWSGlueServiceRole-subra-EZCRC-e3Policy	Customer managed	This policy will be used
<input type="checkbox"/> CloudWatchLogsFullAccess	Customer managed	Provides full access to CloudWatch Logs
<input type="checkbox"/> CloudWatchLogsReadOnlyAccess	Customer managed	Provides read only access to CloudWatch Logs

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

AWS managed

Provides read only access to AWS CloudHSM

AWS managed

Provides full access to AWS Resource Groups and Tag Editor

AWS managed

Provides access to AWS Resource Groups and Tag Editor

AWS managed

Provides access to CloudFront

AWS managed

Provides full access to CloudSearch

AWS managed

Provides read only access to CloudSearch

AWS managed

Provides full access to CloudWatch Metrics

AWS managed

Provides read only access to CloudWatch Metrics

AWS managed

Provides full access to CloudWatch Logs

AWS managed

Provides read only access to CloudWatch Logs

[Cancel](#) [Add permissions](#)

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

#### 4(c): Add configuration

The screenshot shows the AWS Lambda Configuration page. On the left, a sidebar lists options like General configuration, Triggers, Permissions (which is selected), Destinations, Function URL, Environment variables, Tags, VPC, Monitoring and operations tools, and Concurrency. The main area is titled 'Execution role' and shows a role named 'TextExtraction-role-fyuu9sd'. Below this is a 'Resource summary' section for Amazon CloudWatch Logs, indicating 3 actions and 2 resources. A 'View role document' button is present. At the bottom, there's a table for managing resources and actions.

#### 4(d): Add Code

The screenshot shows the AWS Lambda Code source editor. The code editor window displays a file named 'lambda\_function.py' containing Python code for error handling. The code imports sys, traceback, json, and uuid, and defines a process\_error function that formats exceptions into a JSON message. The editor has tabs for Code and Info, and a toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and Upload from.

```
1 import sys
2 import traceback
3 import json
4 import uuid
5 import string
6 from urllib.parse import unquote_plus
7 logger = logging.getLogger()
8 logger.setLevel(logging.INFO)
9
10 def process_error() -> dict:
11     ex_type, ex_value, ex_traceback = sys.exc_info()
12     error_traceback = traceback.format_exception(ex_type, ex_value, ex_traceback)
13     error_msg = json.dumps({
14         "errorType": ex_type.__name__,
15         "errorMessage": str(ex_value),
16         "stackTrace": error_traceback,
17     })
18
19
20
21
22
```

#### Before Adding Trigger

The screenshot shows the AWS Lambda Function overview page for a function named 'TextExtraction'. The left sidebar shows 'Lambda > Functions > TextExtraction'. The main area displays the function name, a thumbnail icon, and a 'Layers' section with '(0)'. Buttons for '+ Add trigger' and '+ Add destination' are visible. To the right, there's a 'Description' field with a note about last modification, a 'Function ARN' field with the value 'arn:aws:lambda:ap-south-1:539979324382:function:TextExtraction', and a 'Function URL' field with a link. Buttons for Throttle, Copy ARN, and Actions are at the top right.

#### 4(e): Now adding trigger

**Trigger configuration** S3

**Bucket**  
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.  
s3/project.awss

**Event types**  
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

**Prefix - optional**  
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.  
e.g. images/

**Suffix - optional**  
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.  
e.g. .jpg

**Recursive invocation**  
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

#### 4(f): After successful adding of trigger.

**TextExtraction**

**Function overview** TextExtraction

**Triggers**

- TextExtraction (S3)

**Actions**

**Description**  
-

**Last modified**  
6 minutes ago

**Function ARN**  
arn:aws:lambda:ap-south-1:539979324382:function:TextExtraction

**Function URL** [Info](#)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

## Step-5: Glue

### 5(a): Create crawler

The screenshot shows the 'Add crawler' wizard in the AWS Glue console. The left sidebar shows various AWS services like EC2, VPC, S3, IAM, Lambda, and AWS Glue. The main area is titled 'Set crawler properties' and shows Step 1: 'Choose data sources and classifiers'. The crawler is named 'projectcrawler' and has a description 'Using Amazon Glue'. The 'Tags - optional' section is collapsed. At the bottom right are 'Cancel' and 'Next' buttons.

### 5(b): Add data source

The screenshot shows the 'Add data source' dialog in the AWS Glue console. The left sidebar shows various AWS services. The dialog is titled 'Add data source' and shows a 'Data source' dropdown set to 'S3'. Below it is a 'Network connection - optional' section with a note about network connections. The 'Location of S3 data' section shows 'In this account' selected. The 'S3 path' field contains 's3://project.aws'. The 'Subsequent crawler runs' section shows 'Crawl all sub-folders' selected. At the bottom right are 'Cancel' and 'Add an S3 data source' buttons.

### 5(c): Choose data sources and classifiers.

The screenshot shows the 'Choose data sources and classifiers' dialog in the AWS Glue console. The left sidebar shows various AWS services. The dialog is titled 'Choose data sources and classifiers' and shows a 'Data source configuration' section with a note about mapped tables. It lists one data source: 's3' with 's3://project.aws' as the data source and 'Recrawl all' as the parameters. The 'Custom classifiers - optional' section is collapsed. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

## 5(d): Configure security settings

The screenshot shows the 'Configure security settings' step of a crawler setup. In the 'IAM role info' section, 'AWSGlueServiceRole-Project' is selected. Below it, there's an optional section for 'Lake Formation configuration' which includes a checkbox for using Lake Formation credentials for crawling S3 data sources. Another optional section for 'Security configuration' is also present. At the bottom, there are 'Cancel', 'Previous', and 'Next' buttons.

## 5(e): Create a database

The screenshot shows the 'Create a database' step. In the 'Database details' section, the 'Name' field is filled with 'projectdb'. There are optional sections for 'Location' and 'Description'. The 'Description' field contains 'Creating database for Glue'. At the bottom right, there is a prominent orange 'Create database' button.

## 5(f): Set output and scheduling.

The screenshot shows the 'Set output and scheduling' step. In the 'Output configuration' section, the 'Target database' is set to 'projectdb'. There are optional fields for 'Table name prefix' and 'Maximum table threshold'. An 'Advanced options' section is also present. At the bottom, there is a 'Crawler schedule' section.

5(g): After successful creation of crawler.

The screenshot shows the AWS Glue service dashboard. In the left sidebar, under 'Data Catalog' > 'Crawlers', the 'projectcrawler' crawler is listed as 'Ready' with a status of 'Succeeded'. The main content area displays a message: 'Crawler successfully starting' and 'The following crawler is now starting: "projectcrawler"'. Below this, it says 'A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.'

5(h): Now open cloud watch logs and check that tables are created or not.

The screenshot shows the CloudWatch Logs interface. The left sidebar shows 'Logs' > 'Log groups' > '/aws-glue/crawlers' > 'projectcrawler'. The main pane displays log events for the 'projectcrawler' crawler. The first event is a 'BENCHMARK : Running Start Crawl for Crawler projectcrawler'. Subsequent events show 'Classification complete, writing results to database projectdb' and 'Crawler configured with Configuration {"Version":1.0,...}'. There are 'Copy' buttons next to each log entry.

5(i): Open tables

The screenshot shows the AWS Glue service dashboard. In the left sidebar, under 'Data Catalog' > 'Tables', three tables are listed: 'output', 'test1.jpg', and 'username.csv'. The main content area displays a message: 'Crawler successfully starting' and 'The following crawler is now starting: "projectcrawler"'. Below this, it says 'A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.'

Name	Database	Location	Classification	Deprecated	Action
output	projectdb	s3://project.aws/output	UNKNOWN	-	Table data
test1.jpg	projectdb	s3://project.aws/test1.j	UNKNOWN	-	Table data
username.csv	projectdb	s3://project.aws/username	csv	-	Table data

Crawler successfully starting  
The following crawler is now starting: "projectcrawler"

AWS Glue > Tables

Tables (3)

Name	Database	Location	Classification	Deprecated	View data
output	projectdb	s3://project.aws/output	UNKNOWN	-	Table data
test1.jpg	projectdb	s3://project.aws/test1.j	UNKNOWN	-	Table data
username_csv	projectdb	s3://project.aws/username	csv	-	Table data

5(j): Now open username.csv table.

AWS Glue > Tables > username\_csv

Last updated (UTC) June 26, 2023 at 10:48:49 Version 0 (Current version) Actions

**Table details**

Name	username_csv	Description	-	Database	projectdb	Classification	csv
Location	s3://project.aws/username.csv	Connection	-	Deprecated	-	Last updated	June 26, 2023 at 10:48:49
Input format	org.apache.hadoop.mapred.TextInPutFormat	Output format	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe	Serde serialization lib	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe		

**Schema**

5(k): This is the schema of the csv file.

AWS Glue > Tables > username\_csv

Last updated (UTC) June 26, 2023 at 10:48:49

**Schema (5)**

#	Column name	Data type	Partition key	Comment
1	username	string	-	-
2	login_email	string	-	-
3	identifier	bigint	-	-
4	first_name	string	-	-
5	last_name	string	-	-

## Step-6: Athena:

### 6(a): Query with Athena

The screenshot shows the Amazon Athena Query editor interface. On the left, the 'Data' sidebar displays the 'Data source' as 'AwsDataCatalog' and the 'Database' as 'projectdb'. Under 'Tables and views', there are three tables listed: 'output', 'test1.jpg', and 'username\_csv'. The 'username\_csv' table has five columns: 'username' (string), 'login\_email' (string), 'identifier' (bigint), 'first\_name' (string), and 'last\_name' (string). The main area shows a query editor with the following SQL code:

```
1 SELECT * FROM "AwsDataCatalog"."projectdb"."username_csv" limit 10;
```

The status bar at the bottom indicates 'No results'.

### 6(b): Now you can also query with s3.

The screenshot shows the Amazon S3 console. In the top navigation bar, it says 'Amazon S3 > Buckets > project.aws > username.csv > Query with S3 Select'. The left sidebar shows 'Buckets' and 'Storage Lens' sections. The main area is titled 'Query with S3 Select' and contains the following details:

- Input settings:** Path: s3://project.aws/username.csv, Size: 241.0 B, Format: CSV (selected), CSV delimiter: Comma (selected).
- SQL query:** A code editor with the following SQL query:

```
1 /* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL query */
2 SELECT * FROM s3object s LIMIT 5
```
- Query results:** Status: Successfully returned 2 records in 289 ms, Bytes returned: 100 B.

The screenshot shows the AWS Glue Query results interface. On the left, there's a sidebar for 'Amazon S3' with options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. Below that are sections for Block Public Access settings and Storage Lens. At the bottom of the sidebar are CloudShell, Feedback, and Language links. The main area is titled 'Query results' and shows a success message: 'Successfully returned 2 records in 289 ms'. It includes a 'Download results' button and tabs for 'Raw' and 'Formatted'. The 'Formatted' tab displays the following data:

Username	Login email	Identifier	First name	Last name
booker12	rachel@example.com	9012	Rachel	Booker

At the bottom right of the main area is a 'Close' button.

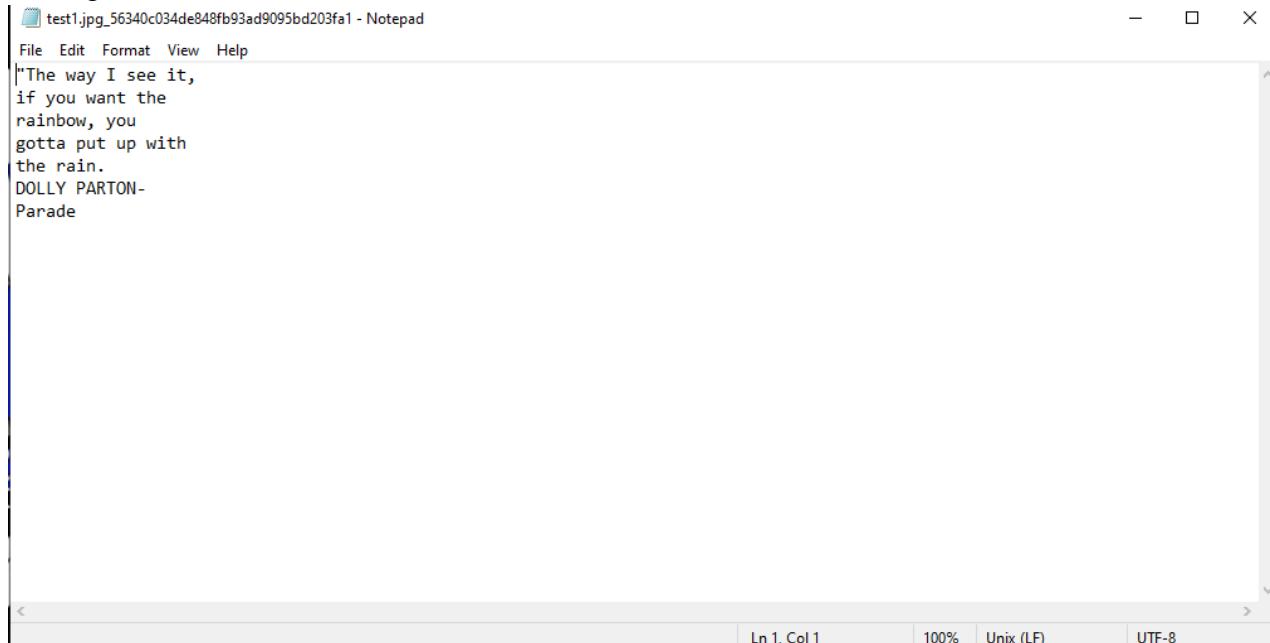
## Step-7: Input file

7(a): Image file1:image file



## Output file1:

And again stored in s3 bucket



A screenshot of a Windows Notepad window titled "test1.jpg\_56340c034de848fb93ad9095bd203fa1 - Notepad". The window contains the following text:

```
"The way I see it,
if you want the
rainbow, you
gotta put up with
the rain.
DOLLY PARTON-
Parade
```

The Notepad window has a standard Windows title bar with icons for minimize, maximize, and close. At the bottom, there are status bars showing "Ln 1, Col 1", "100%", "Unix (LF)", and "UTF-8".

## 7(b): Input file2:csv file

### Output file2:

Username;Login email;Identifier;First name;Last name  
booker12;rachel@example.com;9012;Rachel;Booker  
grey07;;2070;Laura;Grey  
johnson81;;4081;Craig;Johnson  
jenkins46;mary@example.com;9346;Mary;Jenkins  
smith79;jamie@example.com;5079;Jamie;Smith

## Conclusion

As it is used to extract text information from files of any formats it is widely used in detecting typed and hand written text in variety of documents, including financial reports, medical records and tax forms. Extract can extract the data in minutes instead of hours and days, which can be used to quickly automate document processing and act on the information extracted by extracting information from invoices and receipts like automating loans and it has wide applications in real life.