



BLACKBUCKS INTERNSHIP REPORT

LANGUAGE TRANSLATOR AND TEXT TO SPEECH

SUBMITTED BY

TAMMALI AMULYA (21B91A54H0)

PURILLA PALLAVI PRIYA (21B91A6147)

UNDER THE GUIDANCE OF MR. GURU SANTHOSH SEENIVASAN

**Blackbuck Engineers Pvt. Ltd
Road no 36, jubilee hills, Hyderabad**

Title: Language Translator and Text to Speech Convertor.

Abstract:

As some countries conduct the different international business and board meetings in their indigenous languages which is not familiar to other country people. Here the problem raised about language. So we came up with a automated language translator and text to speech convertor.

The main aim of this project is to convert text given in one language into text of another specified language and to convert the text into speech (audio) by using aws services such as Amazon polly, Amazon translate , EC2 and IAM.

Initially create IAM user. create role and policy by giving amazon polly, amazon translate access. add policy to user. Now create EC2 instance. Attach the role to the instance. Connect the instance to terminal and write the html code by connecting to instance region and providing user permissions to instance using access key and other requirements. Now test the project by using public IP of instance.

By using this project we can convert text into required form. This project helps in spreading new information, knowledge and ideas across the world. It also enhances the international marketing and expands the business.

Team members:

Tammali Amulya (21B91A54H0)

Purilla Pallavi Priya (21B91A6147)

TABLE OF CONTENTS

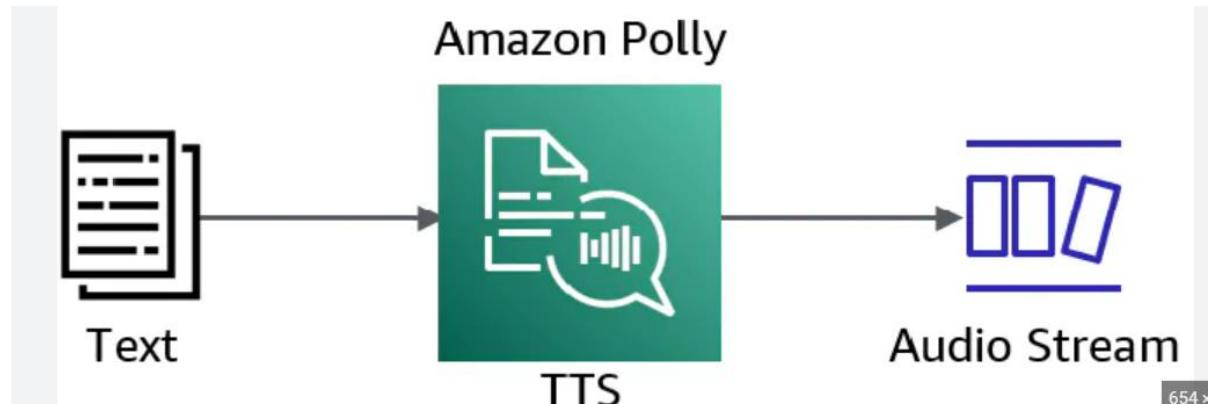
Services used...	4
Rough architecture...	4
Final architecture...	5
Cloud computing...	6
Cloud computing services...	6
IaaS ...	7
PaaS ...	7
SaaS ...	7
Cloud service providers...	8
Amazon web services...	8
Why AWS? ...	9
List of AWS Services...	10
Amazon EC2...	11
Amazon RDS...	12
Multiple AZ Deployment...	13
Read replicas...	13
Performance metrics and monitoring...	13
Amazon VPC...	14
Amazon S3...	14
Amazon IAM...	15
AWS Lambda...	17
AWS Cloud9...	19
Screenshots...	20 – 45

Service used: EC2, IAM, Amazon polly, Amazon Translate.

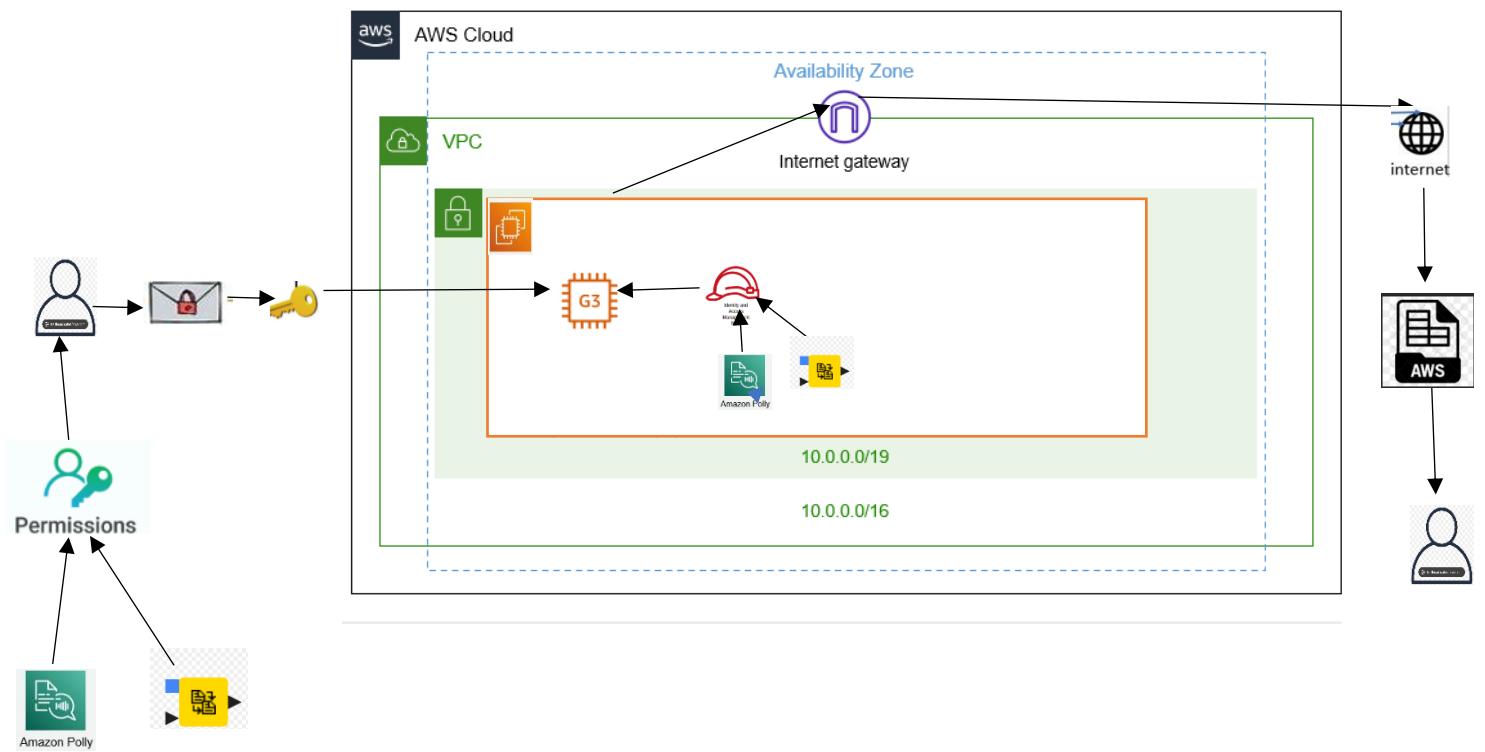
ARCHITECTURES:



Rough architecture for language translator



Rough architecture for text to speech



final architecture of language translator and text to speech

Cloud computing

Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider (or CSP). The CSP makes these resources available for a monthly subscription fee or bills them according to usage.

Compared to traditional on-premises IT, and depending on the cloud services you select, cloud computing helps do the following:

- **Lower IT costs:** Cloud lets you offload some or most of the costs and effort of purchasing, installing, configuring, and managing your own on-premises infrastructure.
- **Improve agility and time-to-value:** With cloud, your organization can start using enterprise applications in minutes, instead of waiting weeks or months for IT to respond to a request, purchase and configure supporting hardware, and install software. Cloud also lets you empower certain users—specifically developers and data scientists.
- **Scale more easily and cost-effectively:** Cloud provides elasticity—instead of purchasing excess capacity that sits unused during slow periods, you can scale capacity up and down in response to spikes and dips in traffic. You can also take advantage of your cloud provider’s global network to spread your applications closer to users around the world. The term ‘cloud computing’ also refers to the technology that makes cloud work. This includes some form of virtualized IT infrastructure—servers, operating system software, networking, and other infrastructure that’s abstracted, using special software, so that it can be pooled and divided irrespective of physical hardware boundaries. For example, a single hardware server can be divided into multiple virtual servers. Cloud Computing Services:

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

are the three most common models of cloud services, and it’s not uncommon for an organization to use all three.

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. 12
- 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 an 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.

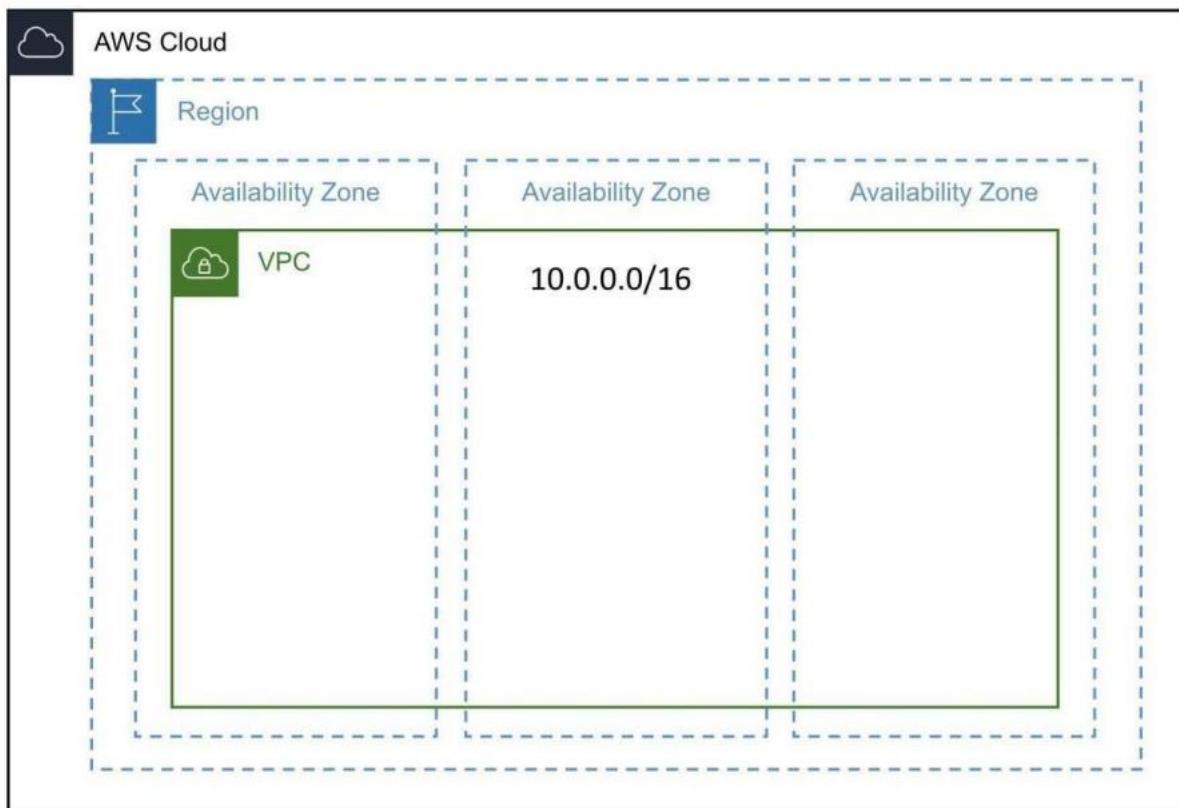
Read replicas allow different use cases such as scale in for read-heavy database workloads. There are up to five replicas available for MySQL, MariaDB, and PostgreSQL. Instances use the native, asynchronous replication functionality of their respective database engines. They have no backups configured 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.

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

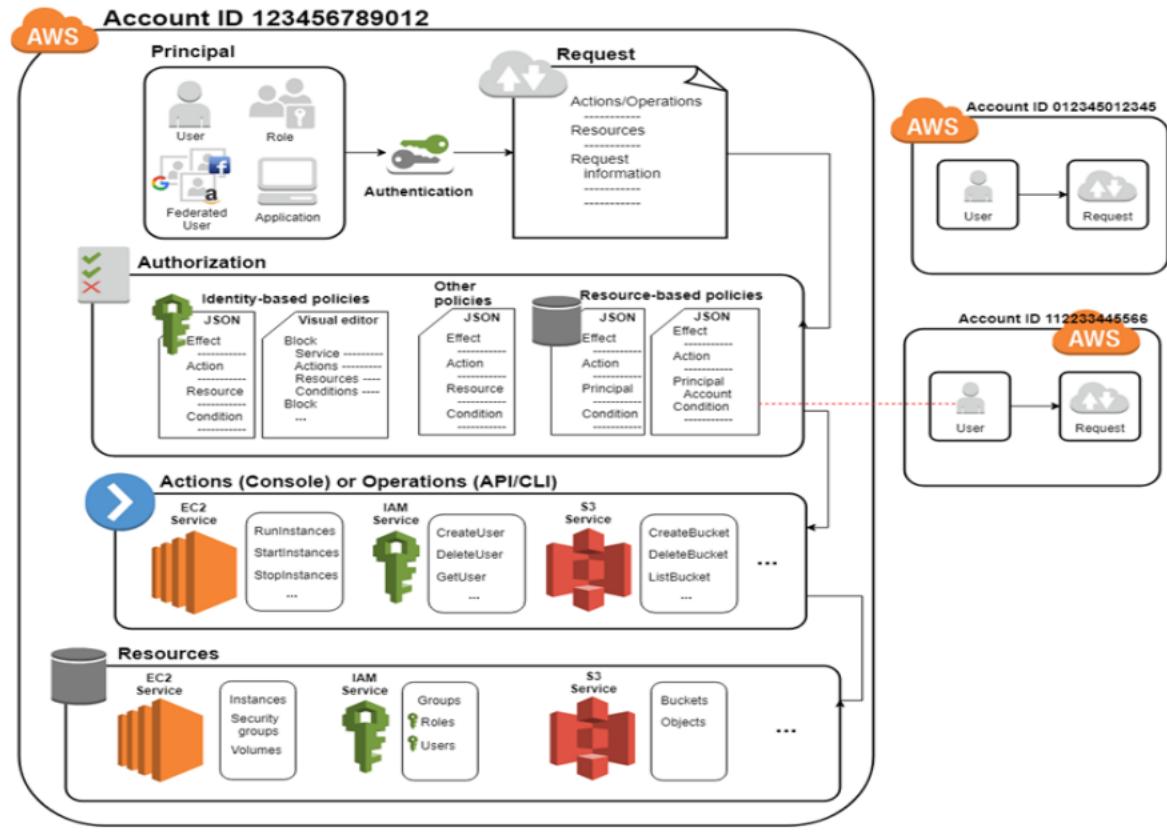
managed using the console provided by Amazon S3, programmatically with the AWS SDK, or the REST application programming interface.

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.



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.

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.

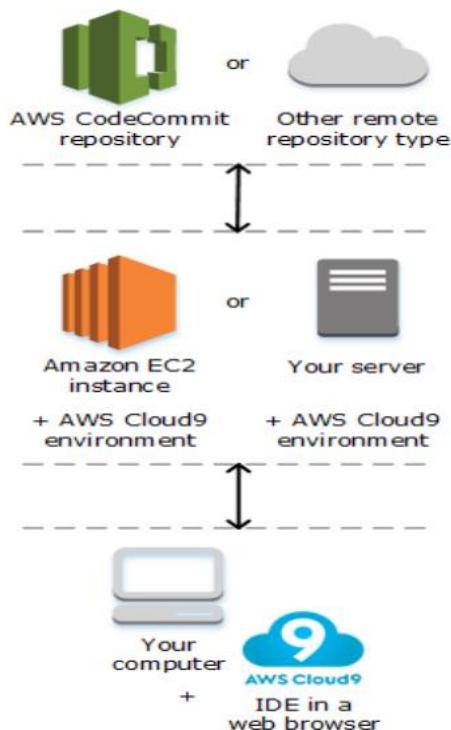


AWS Cloud9:

AWS Cloud9 is an integrated development environment, or IDE.

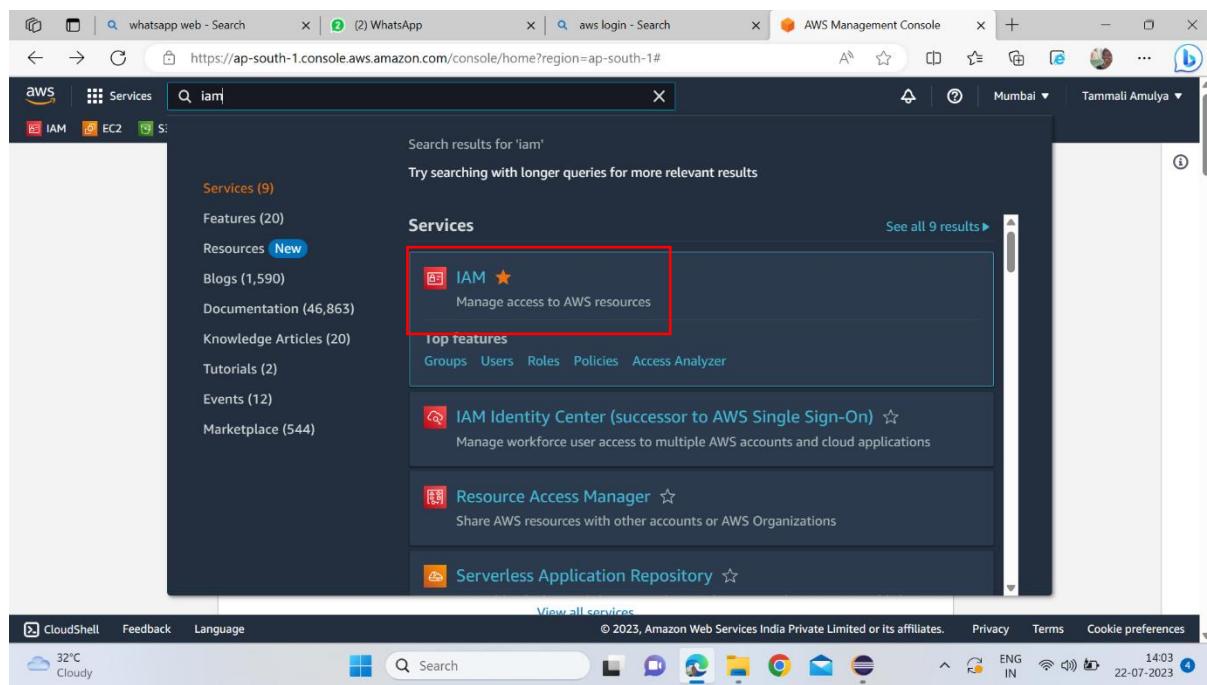
The AWS Cloud9 IDE offers a rich code-editing experience with support for several programming languages and runtime debuggers, and a built-in terminal. It contains a collection of tools that you use to code, build, run, test, and debug software, and helps you release software to the cloud.

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.

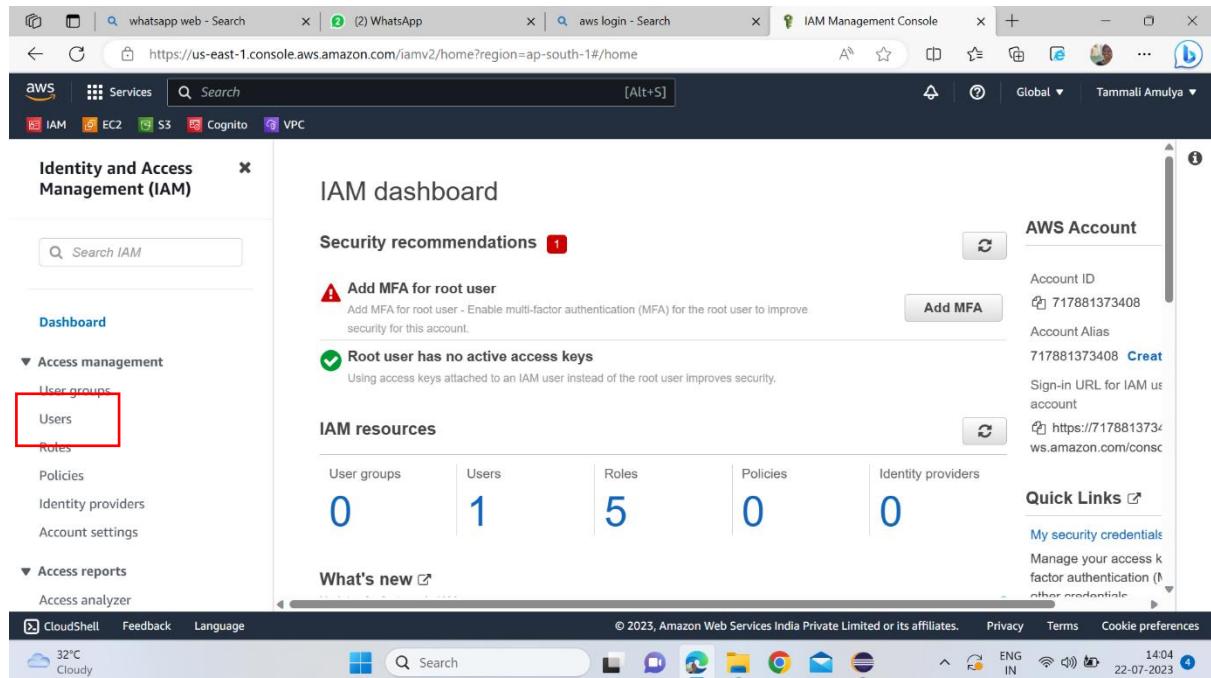


STEPS:

Search for IAM in searchbar and go to IAM service.

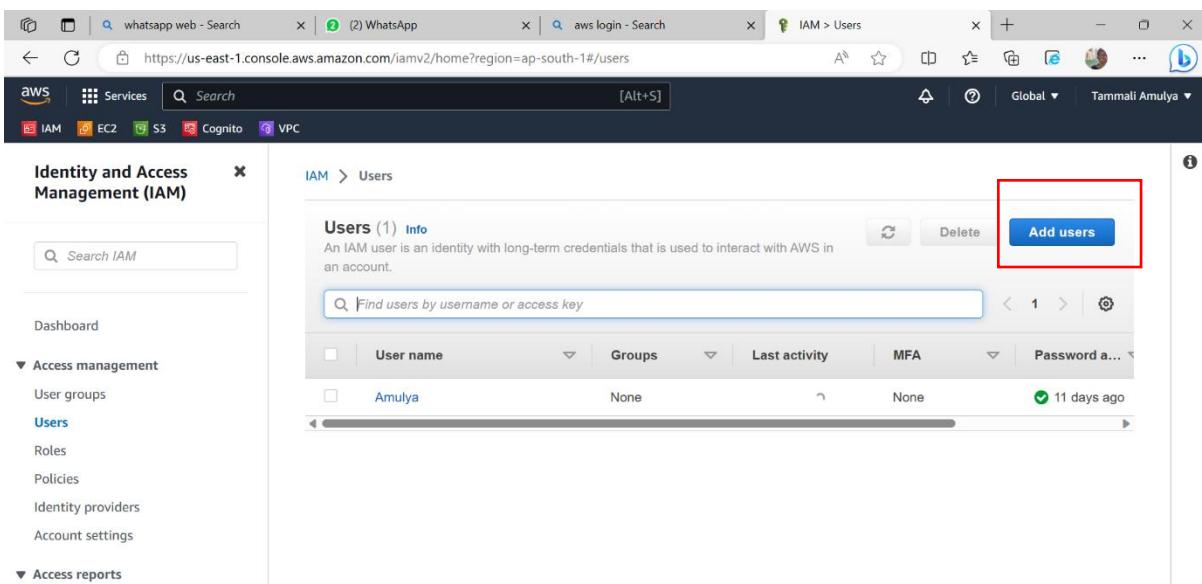


Go to users.



The screenshot shows the AWS IAM Management Console. The left sidebar has a 'User groups' section with a red box around the 'Users' link. The main area displays the IAM dashboard with sections for Security recommendations, IAM resources (User groups: 0, Users: 1, Roles: 5, Policies: 0, Identity providers: 0), and What's new. On the right, there's an AWS Account summary and Quick Links for My security credentials.

Click on add user.



The screenshot shows the AWS IAM Management Console on the 'Users' page. The 'Users (1) Info' section is visible, and the 'Add users' button in the top right corner is highlighted with a red box. A table below lists one user: Amulya, with no groups, last activity 11 days ago, and MFA status None.

Give user name and enable access to AWS management console and enable autogenerated password.

The screenshot shows the AWS IAM 'Create user' wizard at Step 3: Review and create. The 'User name' field is set to 'translator'. The 'Provide user access to the AWS Management Console - optional' checkbox is checked. Below this, a section titled 'Are you providing console access to a person?' has two options: 'Specify a user in Identity Center - Recommended' (unchecked) and 'I want to create an IAM user' (checked). A red arrow points from the 'Autogenerated password' section to the 'Console password' section, which contains the radio button for 'Autogenerated password'.

click on next.

The screenshot shows the AWS IAM 'Create user' wizard at Step 4: Retrieve password. The 'Console password' section shows 'Autogenerated password' selected. A note states: 'You can view the password after you create the user.' Below are two radio button options: 'Custom password' (unchecked) and 'Autogenerated password' (checked). A list of password requirements follows: Must be at least 6 characters long, Must include at least one uppercase letter (A-Z), Must include at least one lowercase letter (a-z), Must include at least one number (0-9), and Must include at least one non-alphanumeric character (! @ # \$ % ^ & * () _ + = [] { } | '). There are also two checkboxes: 'Show password' (unchecked) and 'Users must create a new password at next sign-in - Recommended' (unchecked). A note at the bottom states: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more'.

Enable attach policies directly .

The screenshot shows the AWS IAM 'Create user' wizard at Step 3: Set permissions. In the 'Permissions options' section, the 'Attach policies directly' option is selected and highlighted with a red box. Below this, the 'Permissions policies' section displays a list of 1107 policies, with 'AdministratorAccess' checked and highlighted with a red box.

Give Administrator Access and click on next.

The screenshot shows the AWS IAM 'Create user' wizard at Step 3: Set permissions. A red arrow points to the 'AdministratorAccess' policy in the list of available policies, which is currently selected and highlighted with a blue background.

click on create user.

The screenshot shows two screenshots of the AWS IAM console. The top screenshot is the 'Create user' wizard, Step 4, titled 'Permissions summary'. It shows a single row: 'AdministratorAccess' (Name), 'AWS managed - job function' (Type), and 'Permissions policy' (Used as). Below this is a section for 'Tags - optional' with a note about key-value pairs for identification. The bottom screenshot shows the 'Users' page after a user has been created. A green banner at the top says 'User created successfully'. The main table lists two users: 'Amulya' and 'translator'. The 'Add users' button is highlighted with a red box. The 'translator' row is also highlighted with a red box.

Permissions summary

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Identity and Access Management (IAM)

Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password a...
Amulya	None	11 days ago	None	11 days ago
translator	None	Never	None	None

Add users

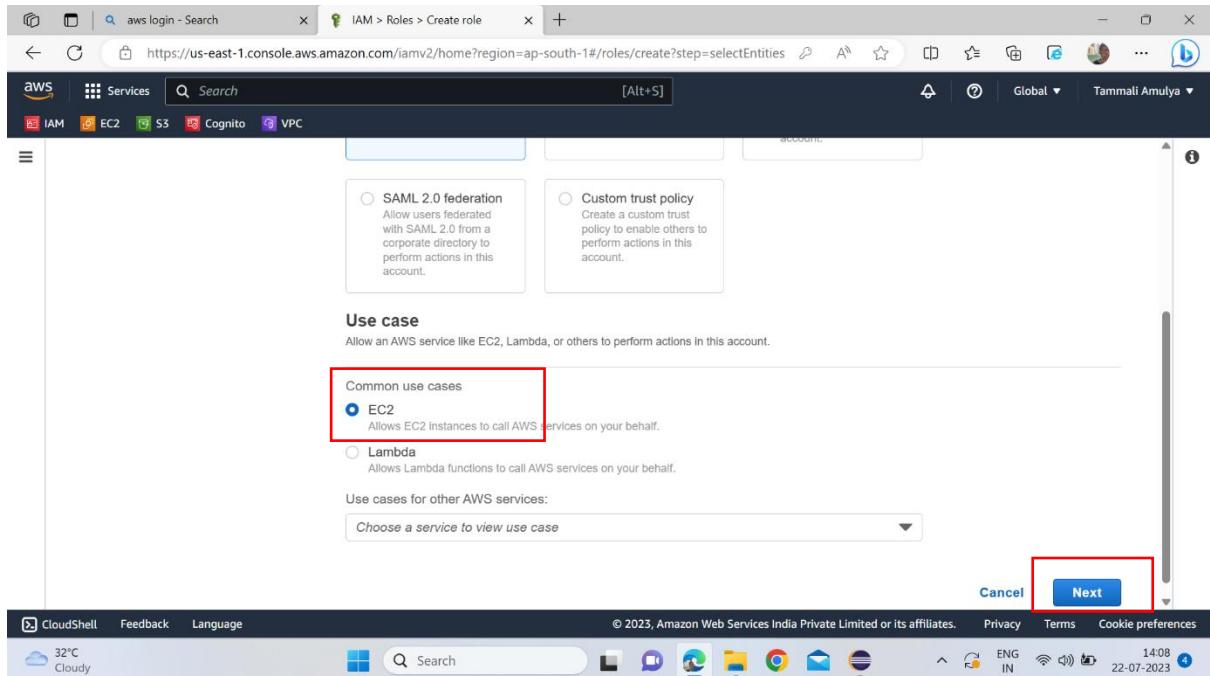
Now go to roles.

The screenshot shows the AWS IAM Management Console dashboard. On the left, there's a sidebar with 'Identity and Access Management (IAM)' at the top, followed by 'Dashboard', 'Access management' (with 'Roles' highlighted and a red box around it), 'Access reports', and 'Access analyzer'. Below these are 'User groups', 'Users', 'Policies', 'Identity providers', and 'Account settings'. The main area is titled 'IAM dashboard' and contains sections for 'Security recommendations' (warning about root user MFA) and 'IAM resources' (showing 1 User, 5 Roles, 0 Policies, and 0 Identity providers). A 'What's new' section is also present. On the right, there's an 'AWS Account' summary with fields like 'Account ID', 'Account Alias', and 'Sign-in URL'. A 'Quick Links' sidebar includes 'My security credentials'. At the bottom, there are standard browser controls and a status bar showing 'CloudShell', 'Feedback', 'Language', 'Cloudy', '32°C', 'ENG IN', '14:04', and '22-07-2023'.

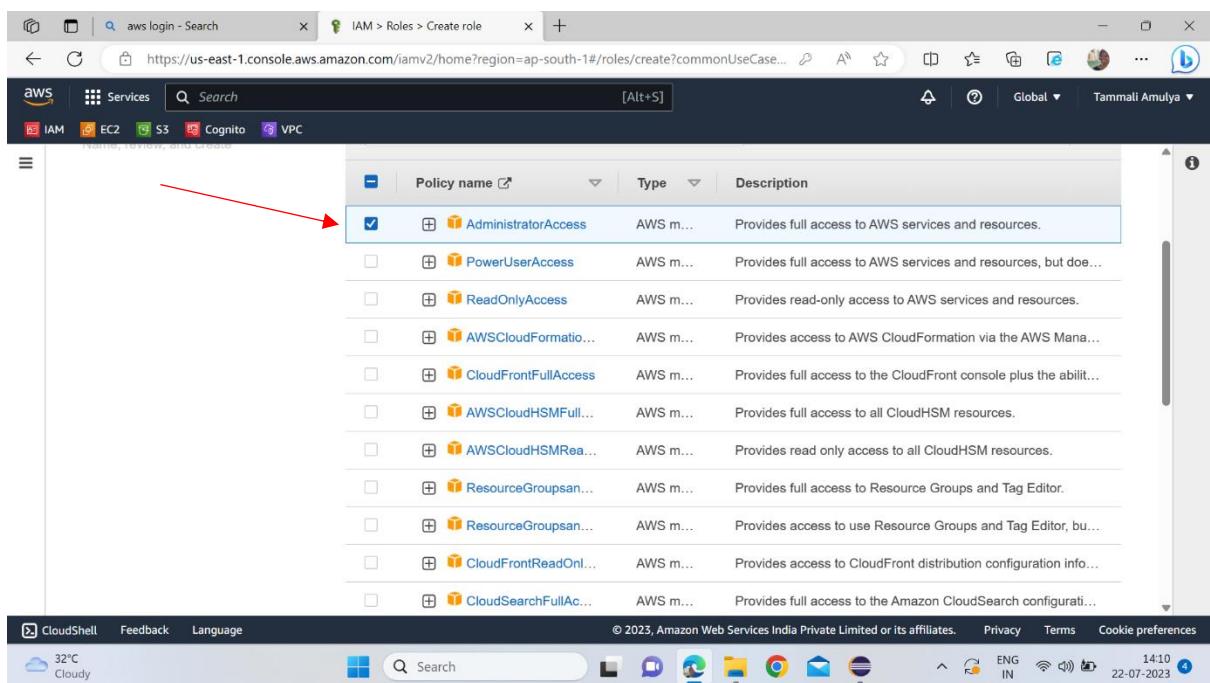
Click on create role.

The screenshot shows the 'IAM > Roles' page. The sidebar on the left is identical to the previous dashboard. The main area is titled 'Roles (5) Info' and describes IAM roles as identities with specific permissions. It lists five existing roles: 'AWSServiceRoleForAutoScaling', 'AWSServiceRoleForElasticLoadBalancing', 'AWSServiceRoleForGlobalAccelerator', 'AWSServiceRoleForSupport', and 'AWSServiceRoleForTrustedAdvisor'. Each role entry includes its name, trusted entity, and last updated time. A red box highlights the 'Create role' button in the top right corner. The bottom of the screen shows the same browser controls and status bar as the previous screenshot.

Enable EC2 service and click on next.



Give administrator access and click on next.



Give role name and description.

The screenshot shows the 'Name, review, and create' step of the IAM Role creation wizard. The 'Role details' section is highlighted with a red box. It contains fields for 'Role name' (set to 'langtransrole') and 'Description' (set to 'Allows EC2 instances to call AWS services on your behalf'). Both fields have character limits and alphanumeric restrictions indicated below them.

Click on create role.

The screenshot shows the final step of the IAM Role creation wizard, 'Permissions policy summary'. It lists a single policy named 'AdministratorAccess' attached as a 'Permissions policy'. Below this, there's a 'Tags' section with an 'Add tag' button and a note about optional tags. At the bottom right, the 'Create role' button is highlighted with a red box.

Our required role is created successfully.

The screenshot shows the AWS IAM Roles page. On the left, the navigation menu is expanded to show 'Access management' with 'Roles' selected. In the main content area, a table lists several service-linked roles. A new role, 'langtransrole', is listed at the bottom of the table and is highlighted with a red border. The table columns include the role name, its purpose, and the last modified date. Below the table, there is a section titled 'Roles Anywhere' with three options: 'Access AWS from your non AWS workloads' (X.509 Standard), 'Operate your non AWS workloads' (Temporary credentials), and 'Authenticate your non AWS workloads and securely provide access to AWS services' (Manage).

Now go to policies.

The screenshot shows the AWS IAM Dashboard. The left sidebar has 'Access management' expanded, with 'Policies' highlighted by a red box. The main area displays the 'IAM dashboard' with sections for 'Security recommendations' (warning about root user MFA) and 'IAM resources'. The 'IAM resources' table shows 0 User groups, 1 User, 5 Roles, 0 Policies, and 0 Identity providers. To the right, there is an 'AWS Account' sidebar with account details like ID and alias, and a 'Quick Links' sidebar with links to security credentials and other IAM features. The status bar at the bottom indicates it's 14:11 on 22-07-2023.

click on create policy.

The screenshot shows the AWS IAM Policies page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and a search bar. The main area has a table titled 'Policies (1105)'. At the top right of the table, there's a 'Actions' dropdown with a 'Create policy' button, which is highlighted with a red box. The table lists various policies like 'AdministratorAccess', 'PowerUserAccess', etc., with columns for 'Policy name', 'Type', 'Used as', and a 'F' icon.

Select translate service and click on next.

The screenshot shows the 'Specify permissions' step in the 'Create policy' wizard. It has two tabs: 'Step 1 Specify permissions' (selected) and 'Step 2 Review and create'. In the main area, there's a 'Policy editor' section with 'Visual', 'JSON', and 'Actions' buttons. Below it, a 'Select a service' dropdown is open, showing 'Translate' highlighted with a red box. There's also a search bar with 'trans' and a 'Popular services' toggle. At the bottom right of the editor, there are 'Cancel' and 'Next' buttons, with 'Next' highlighted with a red box.

Enable Translate Text action.

The screenshot shows the AWS IAM Policies > Create policy interface. In the 'Actions allowed' section, under the 'List' category, there is an unchecked checkbox for 'ListTextTranslationJobs'. Under the 'Read' category, there are two checkboxes: 'DescribeTextTranslationJob' (unchecked) and 'TranslateDocument' (unchecked). Under the 'Write' category, there are two checkboxes: 'StartTextTranslationJob' (unchecked) and 'StopTextTranslationJob' (unchecked). To the right of the 'TranslateDocument' checkbox, there is a red box highlighting the checked checkbox for 'TranslateText'. Below the actions section, there is a 'Resources' section with a note about specifying ARNs. At the bottom of the page, there is a 'Switch to deny permissions' link.

Enable all resources and click on add more permissions.

The screenshot shows the same AWS IAM Policies > Create policy interface as the previous one, but with a red box around the 'Resources' section. Inside the 'Resources' section, there is a radio button for 'Specific' and another for 'All'. A warning message states: '⚠ The alt wildcard ⚠ may be overly permissive for the selected actions. Allowing specific ARNs for these service resources can improve security.' A red arrow points from this warning message down to the 'Add more permissions' button at the bottom of the page. The 'Add more permissions' button is highlighted with a red box.

Select polly service , enable synthesizespeech action and enable all resources and click on next.

The screenshot shows the AWS IAM 'Create policy' interface. At the top, there's a search bar and a breadcrumb trail: 'aws login - Search' > 'IAM > Policies > Create policy'. The main navigation bar includes 'Services' and 'Search' tabs, along with links for 'CloudShell', 'Feedback', and 'Language'. Below the navigation, the 'Polly' service is selected. A red arrow points to the 'Polly' section header. Under 'Actions allowed', a search bar contains 'sp' and a checkbox for 'SynthesizeSpeech' is checked, highlighted by a red box. A link 'Switch to deny permissions' is visible. The 'Resources' section is also highlighted with a red box. The bottom of the screen shows a taskbar with various application icons and system status indicators like weather (31°C Cloudy), time (19:45), and date (23-07-2023).

Give policy name and description.

The screenshot shows the 'Review and create' step of the policy creation wizard. The title 'Step 2 Review and create' is at the top left. To the right, the 'Policy details' section is displayed. It includes fields for 'Policy name' (containing 'pollytranspolicy') and 'Description' (containing 'this policy read source language text and translate it to target language .'). Below this is the 'Permissions defined in this policy' section, which lists the actions defined earlier. A red box highlights the 'Edit' button in this section. The bottom of the screen shows a taskbar with various application icons and system status indicators like weather (32°C Cloudy), time (14:19), and date (22-07-2023).

Click on create policy

The screenshot shows the 'Create policy' step in the AWS IAM console. It displays a table of policy statements:

Service	Access level	Resource	Request conditions
Polly	Full access	All resources	None
Translate	Limited: Read	All resources	None

Below the table, there's a section for 'Add tags - optional' with a note: 'Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.' A 'Add tag' button is present. At the bottom right, there are 'Cancel', 'Previous', and a large blue 'Create policy' button, which is highlighted with a red box.

Required policy is created.

The screenshot shows the 'Policies' page in the AWS IAM console. A green banner at the top indicates: 'Policy pollytranspolicy created.' On the left, a sidebar shows 'Identity and Access Management (IAM)' and 'Access management' sections. The main area lists policies:

Policy name	Type	Used as	Description
pollytranspolicy	Customer managed	None	this policy read source
AdministratorAccess	AWS managed	Permissions policy	Provides full access to all AWS services
PowerUserAccess	AWS managed	None	Provides full access to most AWS services
ReadOnlyAccess	AWS managed	None	Provides read-only access to most AWS services

The 'pollytranspolicy' row is highlighted with a red box. At the bottom right of the table, there's a 'Create policy' button. The browser status bar shows the date and time: '22-07-2023 14:19'.

Now go to users, click on previously created user , go to permissions click on add permissions.

The screenshot shows the AWS IAM 'Users' page. On the left sidebar, 'Users' is selected under 'Access management'. In the main content area, the 'Permissions' tab is active. A red box highlights the 'Permissions policies (1)' section, which lists a single policy named 'AdministratorAccess'. Another red box highlights the 'Add permissions' button at the top right of this section.

Enable attach policy directly.

The screenshot shows the 'Add permissions' wizard, Step 1. It displays three options: 'Add user to group', 'Copy permissions', and 'Attach policies directly'. The 'Attach policies directly' option is selected and highlighted with a red box. Below this, a 'Permissions policies (1107)' section is shown, with a 'Filter by Type' dropdown.

select previously created policy and click on next.

The screenshot shows the 'Add permissions' step of creating a new IAM user. In the 'Permissions policies' section, a policy named 'pollytranspolicy' is selected. The 'Next' button at the bottom right is highlighted with a red box.

Policy name	Type	Attached entities
AmazonPollyFullAccess	AWS managed	0
AmazonPollyReadOnl...	AWS managed	0
pollytranspolicy	Customer managed	0

Click on add permissions.

The screenshot shows the 'Review' step of the IAM User creation process. It displays the user details ('User name: translator') and the attached permissions summary. The 'Add permissions' button at the bottom right is highlighted with a red box.

Name	Type	Used as
pollytranspolicy	Customer managed	Permissions policy

Our policy is attached to user.

The screenshot shows the AWS IAM console with the URL <https://us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/details/translator?section=policies>. A green banner at the top indicates "1 policy added". The main section is titled "Permissions policies (2)" and lists two policies: "AdministratorAccess" (AWS managed - job function, Directly) and "pollytranspolicy" (Customer managed, Directly). A red arrow points from the left margin to the green banner.

Now search for EC2 service and click on EC2.

The screenshot shows the AWS Management Console search results for 'EC'. The search bar at the top has 'EC' typed into it. The results page displays a sidebar with "Services (110)" and a main content area titled "Services". Under "Services", there are three items: "EC2" (Virtual Servers in the Cloud), "Security Hub" (AWS Security Hub is AWS's security and compliance center), and "Security Lake" (Automatically centralize all your security data with a few clicks). A red arrow points from the left margin to the "EC2" item in the search results.

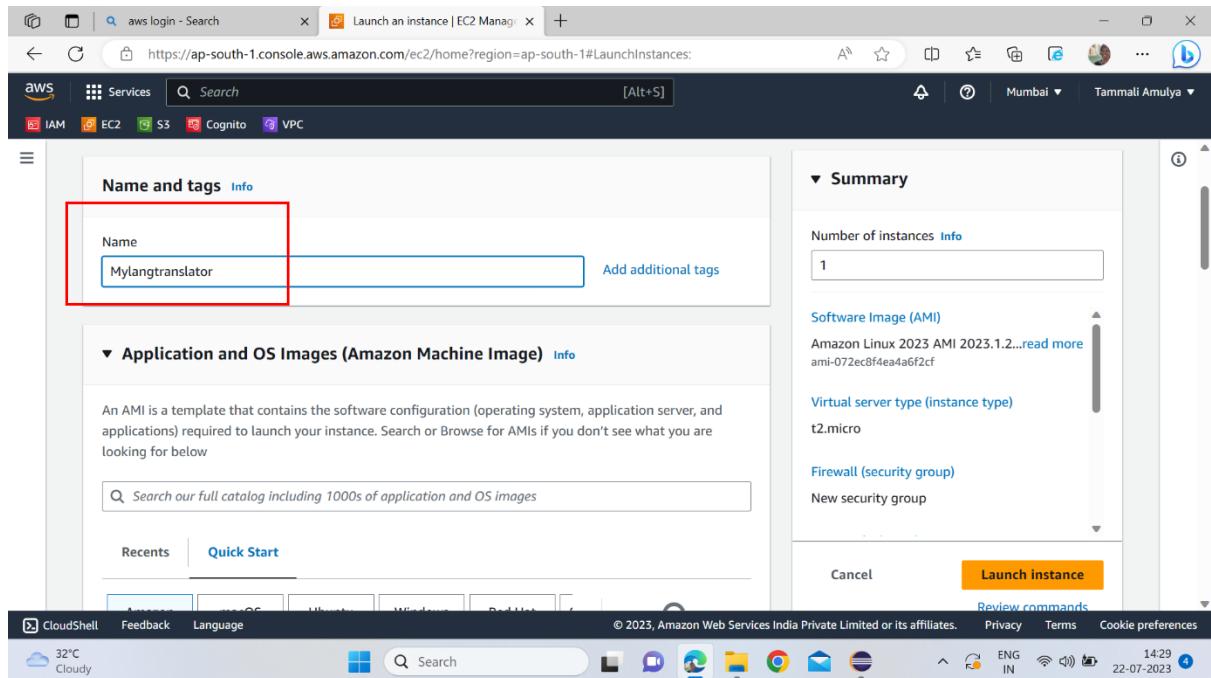
Go to instances.

The screenshot shows the AWS EC2 Management Console. The left sidebar is expanded, showing the 'Instances' section with 'Instances' selected. The main content area displays a summary of resources: 0 running instances, 0 auto scaling groups, 0 dedicated hosts, 0 elastic IPs, 1 instance, 2 key pairs, 0 load balancers, 0 placement groups, 7 security groups, 0 snapshots, and 0 volumes. A tooltip at the bottom of the main content area provides information about easily sizing, configuring, and deploying Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. The top right corner shows account attributes like 'Default VPC' (vpc-0347e63e882289b38) and 'Settings' (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments). The bottom right corner shows system status (32°C Cloudy), search bar, and navigation icons.

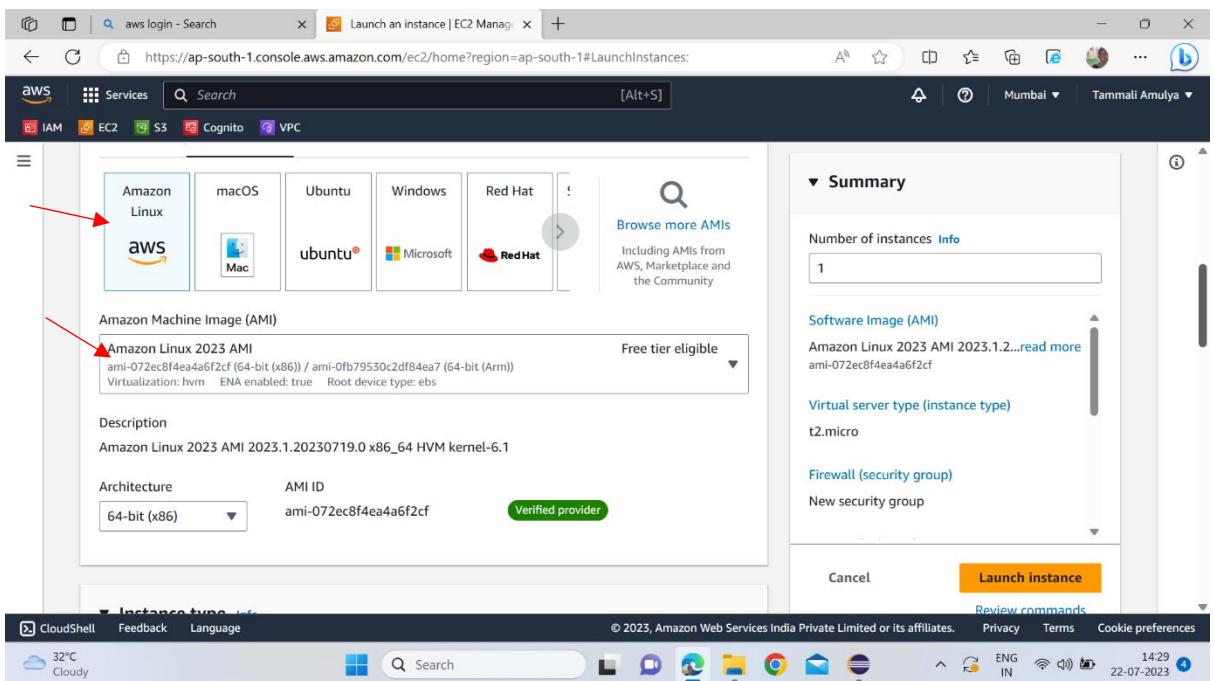
click on launch instances.

The screenshot shows the AWS EC2 Management Console on the 'Instances' page. The 'Actions' dropdown menu has 'Launch instances' selected, which is highlighted with a red box. The main content area shows a table of instances: one named 'Mylangtrans' with Instance ID 'i-0241bd761e092fb76', Instance state 'Terminated', Instance type 't2.micro', and Status check '-'. Below the table, a modal window titled 'Select an instance' is open, prompting the user to choose an instance to launch. The left sidebar is identical to the previous screenshot, showing the 'Instances' section with 'Instances' selected. The bottom right corner shows system status (32°C Cloudy), search bar, and navigation icons.

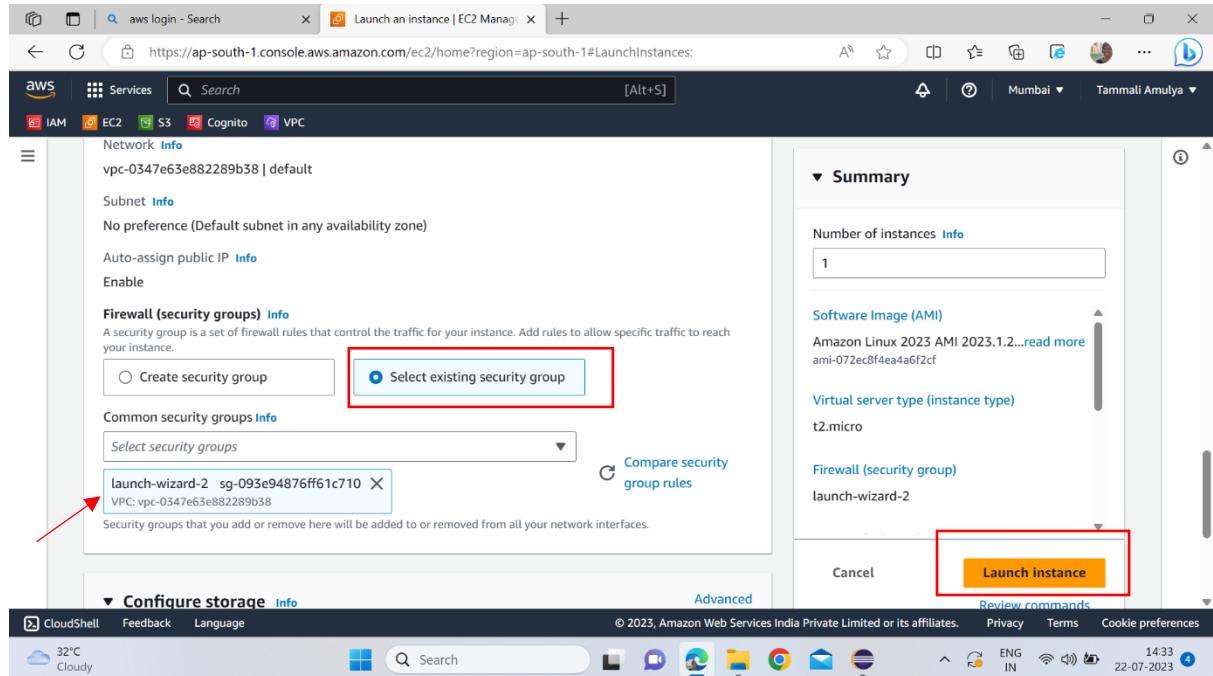
Give instance name.



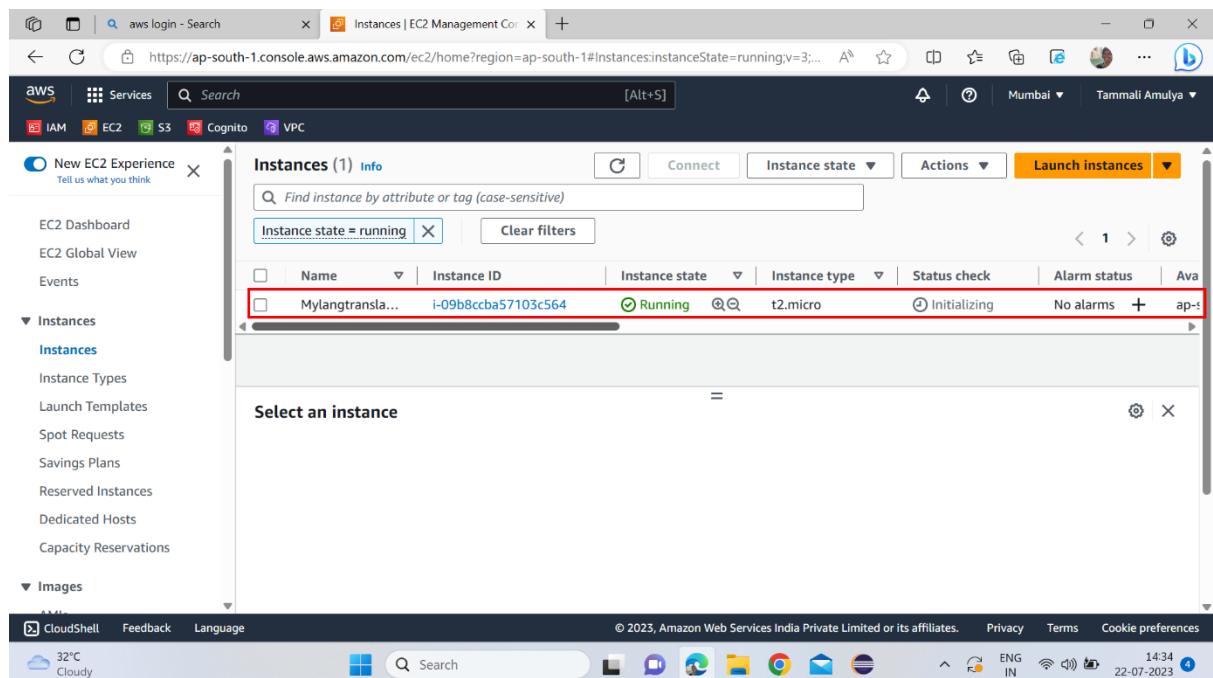
select operating system .



select existing security group and click on launch instances.



required instance created sucessfully



Select instance, click on action, go to security and click on modify IAM role.

The screenshot shows the AWS EC2 Management Console. In the left sidebar, under the 'Instances' section, the 'Instances' link is selected. On the main page, an instance named 'Mylangtransla...' is listed as 'Running'. The 'Actions' dropdown menu is open, and the 'Security' option is highlighted with a red box. Below it, the 'Modify IAM role' button is also highlighted with a red box.

Select previously created role and click on update role.

The screenshot shows the 'Modify IAM role' dialog box. At the top, it says 'Attach an IAM role to your instance.' Below that, the 'Instance ID' is listed as 'i-09b8ccba57103c564 (Mylangtranslator)'. Under the 'IAM role' section, a dropdown menu shows 'langtransrole' selected. At the bottom right of the dialog, the 'Update IAM role' button is highlighted with a red box.

now go to IAM user, go to Security credentials.

The screenshot shows the AWS IAM User Security Credentials page. The URL is https://us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/details/translator?section=sec... . The page has a navigation bar with tabs: Identity and Access Management (IAM), Services, and Global. On the left, there's a sidebar with options like Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports (Access analyzer). The main content area has tabs: Permissions, Groups, Tags, and Security credentials (which is highlighted with a red box). Under Security credentials, there are sections for Console sign-in (Console sign-in link: https://717881373408.sigin.aws.amazon.com/console, Console password updated 32 minutes ago, Last console sign-in: Never) and Multi-factor authentication (MFA) (0). At the bottom, there are buttons for Remove, Resync, and Assign MFA device.

Click on create access key.

The screenshot shows the AWS IAM User Access Keys page. The URL is https://us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/details/translator?section=sec... . The page structure is similar to the previous one, with a sidebar and a main content area. In the main content area, under the Access keys (0) section, there is a button labeled "Create access key" which is highlighted with a red box. Below this, there is a "No access keys" section with a note about best practices and a "Create access key" button. At the bottom, there is a section for SSH public keys for AWS CodeCommit (0) with a "Upload SSH public key" button.

Enable CLI and click on next.

The screenshot shows the AWS IAM 'Create access key' wizard. The current step is 'Access key best practices & alternatives'. A red box highlights the 'Use case' section. Inside, the 'Command Line Interface (CLI)' option is selected, with the sub-note: 'You plan to use this access key to enable the AWS CLI to access your AWS account.' Other options like 'Local code', 'Application running on an AWS compute service', and 'Third-party service' are also listed but not selected.

Access key is created.

The screenshot shows the AWS IAM 'Create access key' wizard. The current step is 'Retrieve access keys'. A red box highlights the 'Access key' section, which displays the generated access key: AKIA2OJIQOLQALBMX6YH. Below it is a 'Secret access key' field with a 'Show' link. The status bar at the bottom indicates 'Access key created'.

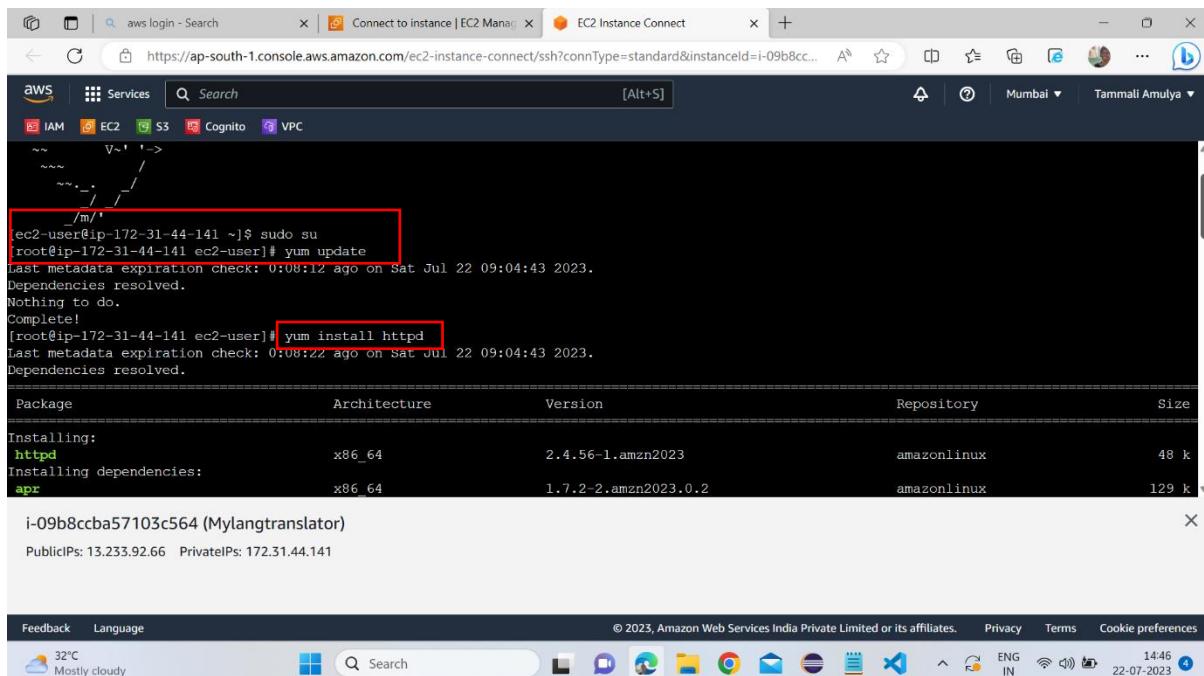
Now go to EC2, select the instance and click on connect.

The screenshot shows the AWS EC2 Management Console. In the left sidebar, under the 'Instances' section, the 'Instances' option is selected. A red arrow points from the 'Instances' label to the 'Instances' section in the sidebar. In the main content area, a table lists one instance: 'Mylangtransla...' with Instance ID 'i-09b8ccba57103c564'. The 'Running' status is indicated by a green circle. To the right of the instance table is a large preview panel for the selected instance. At the top of this panel, there is a 'Connect' button, which is also highlighted with a red box. Below the preview panel are tabs for 'Details', 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'. The 'Details' tab is currently active. At the bottom of the preview panel, there is a note: 'In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.'

Connect the instance.

The screenshot shows the 'Connect to instance' dialog box. At the top, it displays the instance ID 'i-09b8ccba57103c564 (Mylangtranslator)'. Below this, there are two connection type options: 'Connect using EC2 Instance Connect' (selected) and 'Connect using EC2 Instance Connect Endpoint'. The 'Connect using EC2 Instance Connect' option includes a note: 'Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.' The 'Connect using EC2 Instance Connect Endpoint' option includes a note: 'Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.' Further down, there are fields for 'Public IP address' (set to '13.233.92.66') and 'User name' ('ec2-user'). A note at the bottom of the dialog says: 'In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' At the bottom right of the dialog is a large orange 'Connect' button, which is also highlighted with a red box.

Execute the commands such as: sudo su, yum update, yum install httpd,
Service httpd start.



```

aws login - Search | Connect to instance | EC2 Manager | EC2 Instance Connect
https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-09b8cc... | + | Mumbai | Tammali Amulya | ... | b

AWS Services Search [Alt+S]
IAM EC2 S3 Cognito VPC

~~~ V~' '-->
~~~ / \
~~~ / \
~~~ / \
~/'

[ec2-user@ip-172-31-44-141 ~]$ sudo su
[root@ip-172-31-44-141 ec2-user]# yum update
Last metadata expiration check: 0:08:12 ago on Sat Jul 22 09:04:43 2023.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-44-141 ec2-user]# yum install httpd
Last metadata expiration check: 0:08:22 ago on Sat Jul 22 09:04:43 2023.
Dependencies resolved.

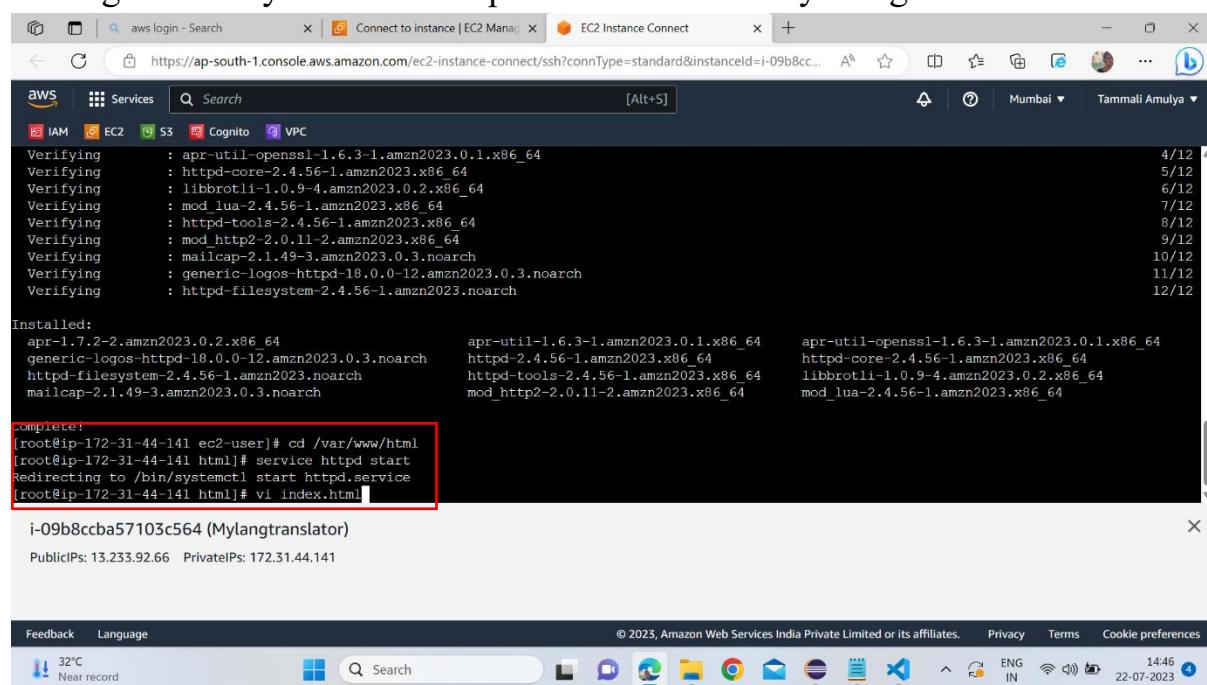
Package           Architecture      Version       Repository      Size
=====
Installing:
httpd             x86_64          2.4.56-1.amzn2023   amazonlinux    48 k
Installing dependencies:
apr               x86_64          1.7.2-2.amzn2023.0.2  amazonlinux  129 k

i-09b8ccba57103c564 (Mylangtranslator)
PublicIPs: 13.233.92.66 PrivateIPs: 172.31.44.141

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
32°C Mostly cloudy Search ENG IN 14:46 22-07-2023 4

```

change directory to html and open index.html file by using commands



```

aws login - Search | Connect to instance | EC2 Manager | EC2 Instance Connect
https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-09b8cc... | + | Mumbai | Tammali Amulya | ... | b

AWS Services Search [Alt+S]
IAM EC2 S3 Cognito VPC

Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
Verifying : httpd-core-2.4.56-1.amzn2023.x86_64
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Verifying : mod_lua-2.4.56-1.amzn2023.x86_64
Verifying : httpd-tools-2.4.56-1.amzn2023.x86_64
Verifying : mod_http2-2.0.11-2.amzn2023.x86_64
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
Verifying : httpd-filesystem-2.4.56-1.amzn2023.noarch

Installed:
apr-util-1.6.3-1.amzn2023.0.1.x86_64
httpd-2.4.56-1.amzn2023.x86_64
httpd-tools-2.4.56-1.amzn2023.x86_64
mod_http2-2.0.11-2.amzn2023.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-filesystem-2.4.56-1.amzn2023.noarch

Complete!
[root@ip-172-31-44-141 ec2-user]# cd /var/www/html
[root@ip-172-31-44-141 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-44-141 html]# vi index.html

i-09b8ccba57103c564 (Mylangtranslator)
PublicIPs: 13.233.92.66 PrivateIPs: 172.31.44.141

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
32°C Near record Search ENG IN 14:46 22-07-2023 4

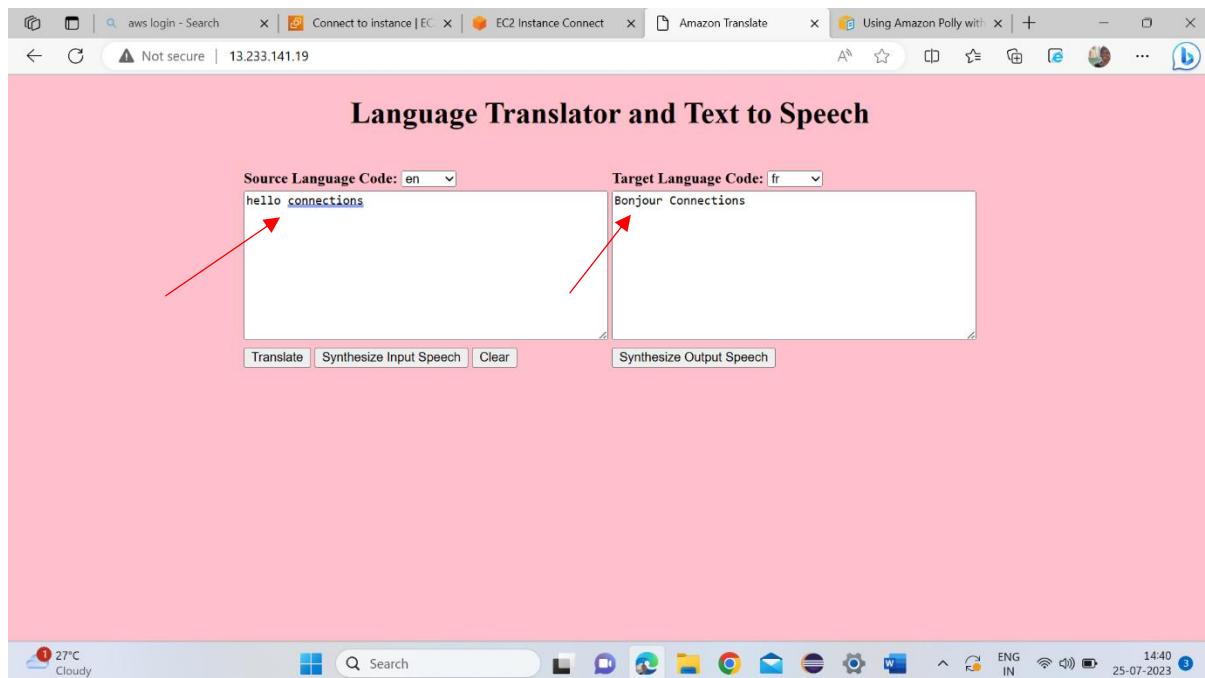
```

Write html with JS code by connecting to our instance region and user with previously created access key.

Save the file and copy the public IP of instance.

```
aws login - Search x | Connect to instance | EC2 Manager x EC2 Instance Connect x +  
https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-09b8cc... A ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌊ ⌋ Mumbai Tammali Amulya  
aws Services Search [Alt+S]  
IAM EC2 S3 Cognito VPC  
Verifying : httpd-core-2.4.56-1.amzn2023.x86_64 5/12  
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 6/12  
Verifying : mod_lua-2.4.56-1.amzn2023.x86_64 7/12  
Verifying : httpd-tools-2.4.56-1.amzn2023.x86_64 8/12  
Verifying : mod_http2-2.0.11-2.amzn2023.x86_64 9/12  
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12  
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 11/12  
Verifying : httpd-filesystem-2.4.56-1.amzn2023.noarch 12/12  
  
Installed:  
apr-1.7.2-2.amzn2023.0.2.x86_64 apr-util-1.6.3-1.amzn2023.0.1.x86_64  
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch httpd-2.4.56-1.amzn2023.x86_64  
httpd-filesystem-2.4.56-1.amzn2023.noarch httpd-tools-2.4.56-1.amzn2023.x86_64  
mailcap-2.1.49-3.amzn2023.0.3.noarch mod_http2-2.0.11-2.amzn2023.x86_64  
  
Complete!  
[root@ip-172-31-44-141 ec2-user]# cd /var/www/html  
[root@ip-172-31-44-141 html]# service httpd start  
Redirecting to /bin/systemctl start httpd.service  
[root@ip-172-31-44-141 html]# vi index.html  
[root@ip-172-31-44-141 html]#  
  
i-09b8ccb Search ... (lyantranslator)  
PublicIPs: 13.233.92.66 PrivateIPs: 172.31.44.141  
  
Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences  
32°C Mostly cloudy ENG IN 14:49 22-07-2023
```

Browse it on a browser and open the index file. Give source text and translate the text into target language and check the speech.



DECLARATION

I hereby declare that “Language translator and text to speech” is the result of the project work carried out by me under the guidance of Mr. Guru Santhosh seenivasan.

DATE:

29/07/2023

SIGNATURE:

T.Amulya, P.Pallavi.