Architecture of the Cloud

Layer 1 (User/Client Layer):

- The users or clients belong to this layer and is the lowest layer in the cloud architecture. From here the client/user initiates the connection to the cloud.
- The client can be any device such as a thin client, thick client, or mobile or any handheld device that would support basic functionalities to access a web application.
- The thin client here refers to a device that is completely dependent on some othersystem for its complete functionality. In simple terms, they have very low processing capability.
- Similarly, thick clients are general computers that have adequate processing capability. They have sufficient capability for independent work.
- Usually, a cloud application can be accessed in the same way as a web application.
- But internally, the properties of cloud applications are significantly different.Thus, this layer consists of client devices

Layers in theArchitecture of the Cloud

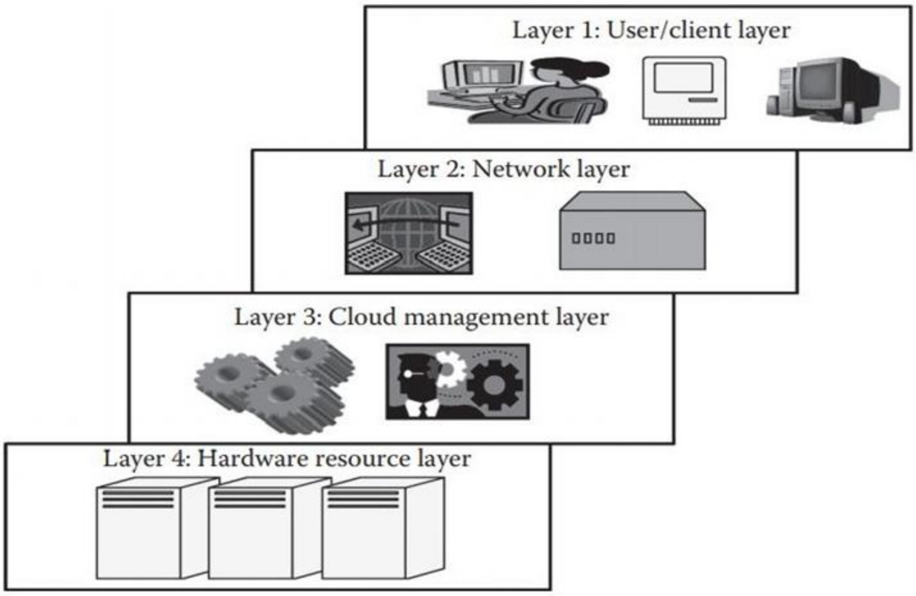

Layer 1: User/client layer
Layer 2: Network layer
Layer 3: Cloud management layer
Layer 4: Hardware resource layer

FIGURE 3.1
Cloud architecture.

Layer 2 (Network Layer):

- This layer allows the users to connect to the cloud.
- The whole cloud infrastructure is dependent on this connection where the services are offered to the customers.
- This is primarily the Internet in the case of a public cloud.
- The public cloud usually exists in a specific location and the user would not know the location as it is abstract.
- And, the public cloud can be accessed all over the world.
- In the case of a private cloud, the connectivity may be provided by a local area network (LAN).
- Even in this case, the cloud completely depends on the network that is used.
- Usually, when accessing the public or private cloud, the users require minimumbandwidth, which is sometimes defined by the cloud providers.
- This layer does not come under the purview of service-level agreements (SLAs), that is,SLAs do not take into account the Internet connection between the user and cloud for quality of service (QoS).

Layer 3 (Cloud Management Layer):

- This layer consists of softwares that are used in managing the cloud.
- The softwares can be a cloud operating system (OS), a software that acts as an interface between the data center (actual resources) and the user, or a management software that allows managing resources.
- These softwares usually allow resource management (scheduling, provisioning, etc.), optimization (server consolidation, storage workload consolidation), and internal cloud governance.
- This layer comes under the purview of SLAs, that is, the operations taking place in this layer would affect the SLAs that are being decided upon between the users and the service providers.
- Any delay in processing or any discrepancy in service provisioning may lead to an SLA violation.
- As per rules, any SLA violation would result in a penalty to be given by the service provider.
- These SLAs are for both private and public clouds Popular service providers areAmazon Web Services (AWS) and Microsoft Azure

for public cloud.

- Similarly, OpenStack and Eucalyptus allow private cloud creation, deployment, and management.

Layer 4 (Hardware Resource Layer):

- Layer 4 consists of provisions for actual hardware resources.
- Usually, in the case of a public cloud, a data center is used in the back end.
- Similarly, in a private cloud, it can be a data center, which is a huge collection of hardware resources interconnected to each other that is present in a specific location or a high configuration system.
- This layer comes under the purview of SLAs.
- This is the most important layer that governs the SLAs.
- This layer affects the SLAs most in the case of data centers.
- Whenever a user accesses the cloud, it should be available to the users as quickly as possible and should be within the time that is defined by the SLAs.
- As mentioned, if there is any discrepancy in provisioning the resources or application, the service provider has to pay the penalty.
- Hence, the data center consists of a high-speed network connection and a highlyefficient algorithm to transfer the data from the data center to the manager.
- There can be a number of data centers for a cloud, and similarly, a number of clouds can share a data center.

A.What is cloud management?

- Cloud management is the control and oversight of an organization's infrastructure, services, and applications that run in the cloud.
- More and more organizations are moving their IT setup into the cloud to enjoy the flexibility, scale, and cost benefits that this technology brings.
- However, all cloud resources must be configured and managed systematically for optimum security, efficiency, and cost control.
- Cloud management includes policies, strategies, and technologies to control and maintain private, public, and hybrid cloud resources.

B.Benefits of cloud management

- Cloud management gives organizations a single point of control over the vast resources they deploy on the cloud.
- Organizations can balance innovative cloud expansion with governance, cost control, and flexibility.
- We share several benefits that organizations will gain from using cloud management solutions.

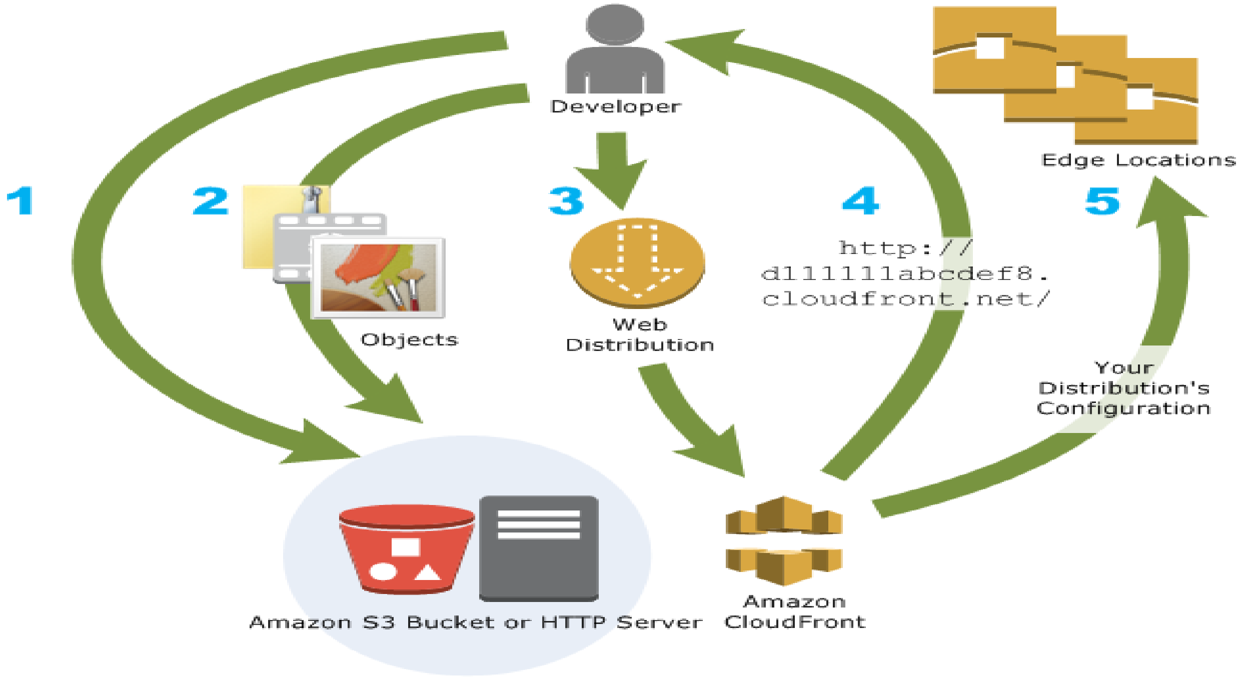
1) **Ease of use**
    - Cloud management tools provide user-friendly interfaces that allow system administrators to deploy, manage, and scale resources across single or multi-cloud environments.
    - Instead of individually provisioning resources for cloud workloads, IT teams can deploy several quickly from the cloud management platform in simple steps.
    - Moreover, cloud management tools enable continuous monitoring, so all events and alerts are channeled to the respective departments.
2) **Centralized cloud governance**
    - Organizations migrating to the cloud need a robust governance and compliance strategy to streamline policy implementation and access control.
    - With a cloud management platform, organizations can enforce consistent business and security policies on applications, services, databases, and other workloads they deploy on the cloud.

- This reduces operating overheads in conventional setups where organizations spend considerable resources to manage disparate business workflows.

3) **Cost and capacity control**
    - Cloud sprawl refers to deploying cloud resources without centralized control and accounting.
    - Organizations can handle cloud sprawl more effectively with cloud management solutions.
    - Administrators can keep track of unused cloud resources and reallocate or shut them down appropriately.
4) **Automated incident response**
    - Cloud workloads may occasionally experience incidents that require manual intervention.
    - Organizations implement cloud management strategies to automate incident management and accelerate disaster recovery.
    - They can improve service availability, operational reliability, and customer satisfaction.
5) **Multi-cloud management**
    - Scaling digital resources across multiple cloud environments requires navigating the infrastructural differences between public, private, and hybrid clouds.
    - A cloud management platform provides organizations with the necessary resources and software tools to bridge the infrastructural gap between different cloud setups.
    - For example, you can move a general knowledge base to the public cloud while retaining proprietary data in the private cloud network.

- **Amazon CloudFront**
- AWS CloudFront is a **content delivery network (CDN)** service provided by Amazon Web Services.
- It helps deliver content, such as web pages, videos, images, and other assets, to users with **low latency** and **high transfer speeds**.
- CloudFront works by caching copies of your content in multiple locations around the world, called **edge locations**.
- When a user requests content, CloudFront serves it from the nearest edge location, reducing the distance the data must travel and improving performance.
- If the content is already in the edge location with the **lowest latency**, CloudFront delivers it immediately.
- If the content is not in that edge location, CloudFront retrieves it from an **origin** that you've defined—such as an Amazon S3 bucket, a MediaPackage channel, or an HTTP server (for example, a web server)
- You also get increased **reliability** and **availability** because copies of your files (also known as objects) are now held (or cached) in multiple edge locations around the world.

How you configure CloudFront to deliver your content


Developer
Edge Locations
1
2
3
4
5
Objects
Web Distribution
http://d111111abcdef8.cloudfront.net/
Your Distribution's Configuration
Amazon S3 Bucket or HTTP Server
Amazon CloudFront

Step1

- You **specify origin servers** which stores the original, definitive version of your objects.
- If you're serving content over HTTP, your origin server is either an Amazon S3 bucket or an HTTP server, such as a web server.
- Your HTTP server can run on an Amazon Elastic Compute Cloud (Amazon EC2) instance or on a server that you manage; these servers are also known as **custom origins.**
- From these servers CloudFront gets your files which will then be distributed from CloudFront edge locations all over the world.

Step2

- You **upload your files** to your origin servers. Your files, also known as objects, typically include web pages, images, and media files, but can be anything that can be served over HTTP.
- If you're using an Amazon S3 bucket as an origin server, you can make the objects in your bucket publicly readable, so that anyone who knows the CloudFront URLs for your objects can access them.
- You also have the option of keeping objects private and controlling who accesses them.

Step3

- You **create a CloudFront distribution**, which tells CloudFront which origin servers to get your files from when users request the files through your web site or application.
- At the same time, you specify details such as whether you want CloudFront to log all requests and whether you want the distribution to be enabled as soon as it's created.

Step4

- **CloudFront assigns a domain name** to your new distribution that you can see in the CloudFront console, or that is returned in the response to a programmatic request, for example, an API request.
- If you like, you can add an alternate domain name to use instead.

Step5

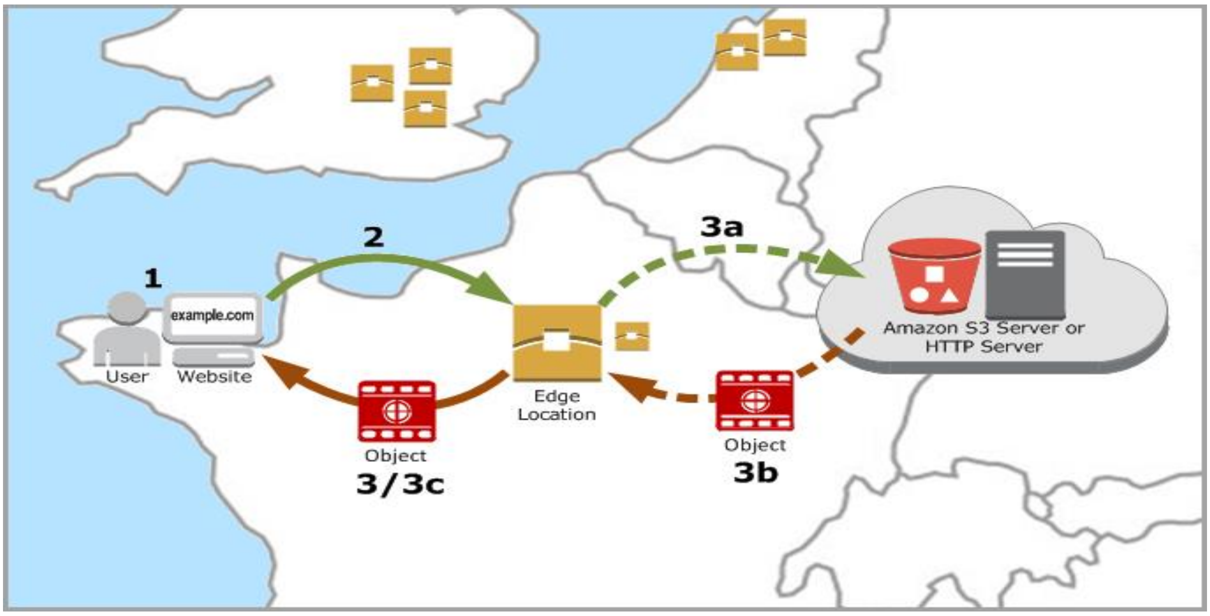
- CloudFront sends your distribution's configuration (but not your content) to all of its edge locations or points of presence (POPs)—collections of servers in geographically-dispersed data centers where CloudFront caches copies of your files.

As you develop your website or application, you use the domain name that CloudFront provides for your URLs. For example, if CloudFront returns d111111abcdef8.cloudfront.net as the domain name for your distribution, the URL for logo.jpg in your Amazon S3 bucket (or in the root directory on an HTTP server) is https://d111111abcdef8.cloudfront.net/logo.jpg.

Or you can set up CloudFront to use your own domain name with your distribution. In that case, the URL might be https://www.example.com/logo.jpg.

Optionally, you can configure your origin server to add headers to the files, to indicate how long you want the files to stay in the cache in CloudFront edge locations. By default, each file stays in an edge location for 24 hours before it expires. The minimum expiration time is 0 seconds; there isn't a maximum expiration time.

How CloudFront delivers content to your users



After you configure CloudFront to deliver your content, here’s what happens when users request your objects:

1. A user accesses your website or application and sends a request for an object, such as an image file or an HTML file.
2. DNS routes the request to the CloudFront POP (edge location) that can best serve the request, typically the nearest CloudFront POP in terms of latency.

3. CloudFront checks its cache for the requested object. If the object is in the cache, CloudFront returns it to the user. If the object is *not* in the cache, CloudFront does the following:

a. CloudFront compares the request with the specifications in your distribution and forwards the request to your origin server for the corresponding object—for example, to your Amazon S3 bucket or your HTTP server.

b. The origin server sends the object back to the edge location.

c. As soon as the first byte arrives from the origin, CloudFront begins to forward the object to the user. CloudFront also adds the object to the cache for the next time someone requests it.

What is AWS billing service?

- AWS Billing and Cost Management is a **web service** that provides features that helps you pay your bills and optimize your costs.
- Amazon Web Services bills your account for usage, which ensures that you pay only for what you use.
- AWS Billing and Cost Management provides a suite of features to help you set up your billing, retrieve and pay invoices, and analyze, organize, plan, and optimize your costs.
- To get started, set up your billing to match your requirements. For individuals or small organizations, AWS will automatically charge the credit card provided.
- For larger organizations, you can use AWS Organizations to consolidate your charges across multiple AWS accounts.

Features of AWS Billing

1) Billing and payments

Understand your monthly charges, view and pay invoices, and manage preferences for billing, invoices, tax, and payments.

- **Bills page** – Download invoices and view detailed monthly billing data to understand how your charges were calculated.
- **Purchase orders** – Create and manage your purchase orders to comply with your organization’s unique procurement processes.
- **Payments** – Understand your outstanding or past-due payment balance and payment history.
- **Payment profiles** – Set up multiple payment methods for different AWS service providers or parts of your organization.
- **Credits** – Review credit balances and choose where credits should be applied.
- **Billing preferences** – Enable invoice delivery by email and your preferences for credit sharing, alerts, and discount sharing**.**

2) Cost analysis

Analyze your costs, export detailed cost and usage data, and forecast your spending.

- **AWS Cost Explorer** – Analyze your cost and usage data with visuals, filtering, and grouping. You can forecast your costs and create custom reports.
- **Data exports** – Create custom data exports from Billing and Cost Management datasets.
- **Cost Anomaly Detection** – Set up automated alerts when AWS detects a cost anomaly to reduce unexpected costs.
- **AWS Free Tier** – Monitor current and forecasted usage of free tier services to avoid unexpected costs.

3) Cost organization

Organize your costs across teams, applications, or end customers.

- **Cost categories** – Map costs to teams, applications, or environments, and then view costs along these dimensions in Cost Explorer and data exports. Define split charge rules to allocate shared costs.
- **Cost allocation tags** – Use resource tags to organize, and then view costs by cost allocation tag in Cost Explorer and data exports.

4) Budgeting and planning

Estimate the cost of a planned workload, and create budgets to track and control costs.

- **Budgets** – Set custom budgets for cost and usage to govern costs across your organization and receive alerts when costs exceed your defined thresholds.

- **Savings and commitments**
- Optimize resource usage and use flexible pricing models to lower your bill.

- **AWS Cost Optimization Hub** – Identify savings opportunities with tailored recommendations including deleting unused resources, rightsizing, Savings Plans, and reservations.

- **Savings Plans** – Reduce your bill compared to on-demand prices with flexible pricing models. Manage your Savings Plans inventory, review purchase recommendations, and analyze Savings Plan utilization and coverage.

- **Reservations** – Reserve capacity at discounted rates for Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS), Amazon Redshift, Amazon DynamoDB, and more.

What is IAM?

- AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources.
- With IAM, you can centrally manage permissions that control which AWS resources users can access.
- You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

IAM features

IAM gives you the following features:

1.Shared access to your AWS account

- You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key.

2.Granular permissions

- You can grant different permissions to different people for different resources.
- For example, you might allow some users complete access to Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Redshift, and other AWS services.
- For other users, you can allow read-only access to just some S3 buckets, or permission to administer just some EC2 instances, or to access your billing information but nothing else.

3.Secure access to AWS resources for applications that run on Amazon EC2

- You can use IAM features to securely provide credentials for applications that run on EC2 instances. These credentials provide permissions for your application to access other AWS resources. Examples include S3 buckets and DynamoDB tables.

4.Multi-factor authentication (MFA)

- You can add two-factor authentication to your account and to individual users for extra security.
- With MFA you or your users must provide not only a password or access key to work with your account, but also a code from a specially configured device.

Accessing IAM

You can work with AWS Identity and Access Management in any of the following ways.

AWS Management Console

- The console is a browser-based interface to manage IAM and AWS resources.

AWS Command Line Tools

- You can use the AWS command line tools to issue commands at your system's command line to perform IAM and AWS tasks.
- Using the command line can be faster and more convenient than the console.
- The command line tools are also useful if you want to build scripts that perform AWS tasks.
- AWS provides two sets of command line tools: the **AWS Command Line Interface (AWS CLI)** and **the AWS Tools for Windows PowerShell.**

AWS SDKs

- AWS provides SDKs (software development kits) that consist of libraries and sample code for various programming languages and platforms (Java, Python, Ruby, .NET, iOS, Android, etc.).
- The SDKs provide a convenient way to create programmatic access to IAM and AWS.
- For example, the SDKs take care of tasks such as cryptographically signing requests, managing errors, and retrying requests automatically

IAM Query API

- You can access IAM and AWS programmatically by using the IAM Query API, which lets you issue HTTPS requests directly to the service.
- When you use the Query API, you must include code to digitally sign requests using your credentials.

Creating an IAM user in your AWS account

The process of creating a user and enabling that user to perform work tasks consists of the following steps:

1) **Create the user** in the AWS Management Console, the AWS CLI, Tools for Windows PowerShell, or using an AWS API operation.
2) **Create credentials** for the user, depending on the type of access the user requires
3) Give the user **permissions** to perform the required tasks by adding the user to one or more groups. You can also grant permissions by attaching permissions policies directly to the user. You can also use a permissions boundary to limit the permissions that a user can have, though this is not common.
4) (Optional) **Add metadata** to the user by attaching tags.
5) Provide the user with **the necessary sign-in information**. This includes the password and the console URL for the account sign-in page where the user provides those credentials.
6) (Optional) Configure **multi-factor authentication (MFA)** for the user. MFA requires the user to provide a one-time-use code each time he or she signs into the AWS Management Console.
7) (Optional) Give users permissions to manage their **own security** credentials. (By default, users do not have permissions to manage their own credentials.)

User creation can be done in 3 ways

- **A.** Creating IAM users (**console**)
- **B.** Creating IAM users (**AWS CLI**)
- **C.** Creating IAM users (**AWS API**)

A.Creating IAM users (console)

1. To create an IAM using AWS Management **console**, sign-in Aws account.
2. On the Console Home page, select the **IAM service**.
3. In the navigation pane, select **Users** and then select **Add users**.

4. On the Specify user details page, under **User details**, in **User name**, enter the name for the new user. This is their sign-in name for AWS.
5. Select **Provide user access to the – AWS Management Console optional** This produces AWS Management Console sign-in credentials for the new user. You are asked whether **are you providing console access to a person.**
    - select **I want to create an IAM user** and continue following this procedure.

   a) For Console password, select one of the following:

      Autogenerated password – The user gets a randomly generated password that meets the account password policy. You can view or download the password when you get to the Retrieve password page.
      Custom password – The user is assigned the password that you enter in the box.

   b) (Optional) **Users must create a new password at next sign-in (recommended)** is selected by default to ensure that the user is forced to change their password the first time they sign in.
6. Select **Next.**
7. On the **Set permissions page**, specify how you want to assign permissions for this user. Select one of the following three options:

    - **Add user to group** – Select this option if you want to assign the user to one or more groups that already have permissions policies. IAM displays a list of the groups in your account, along with their attached policies.

    - **Copy permissions** – Select this option to copy all of the group memberships, attached managed policies, embedded inline policies, and any existing permissions boundaries from an existing user to the new user. IAM displays a list of the users in your account. Select the one whose permissions most closely match the needs of your new user.

    - **Attach policies directly** – Select this option to see a list of the AWS managed and customer managed policies in your account. Select the policies that you want to attach to the user or select

Create policy to open a new browser tab and create a new policy. After you create the policy, close that tab and return to your original tab to add the policy to the user.policies. After you create the policy, close that tab and return to your original tab to add the policy to the user.

8. (Optional) Set a permissions boundary. This is an advanced feature. Open the **Permissions boundary** section and select **Use a permissions boundary to control the maximum permissions**. IAM displays a list of the AWS managed and customer managed policies in your account. Select the policy to use for the permissions boundary or select Create policy After you create the policy, close that tab and return to your original tab to select the policy to use for the permissions boundary.
9. Select **Next.**
10. (Optional) On the **Review and create page**, under **Tags**, select **Add new tag** to add metadata to the user by attaching tags as key-value pairs.
11. Review all of the choices you made up to this point. When you are ready to proceed, select **Create user**.
12. On the **Retrieve password** page, get the password assigned to the user:
    - Select **Show** next to the password to view the user's password so that you can record it manually.
    - Select **Download .csv** to download the user's sign in credentials as a .csv file that you can save to a safe location.
13. Select **Email sign-in instructions**. Your local mail client opens with a draft that you can customize and send to the user. The email template includes the following details to each user:
    - User name
    - URL to the account sign-in page. Use the following example, substituting the correct account ID number or account alias:
        - https://AWS-account-ID or alias.signin.aws.amazon.com/console

B.Creating IAM users(AWS CLI)

Steps to create an IAM user using AWS CLI

1. Create a user.
    - aws iam create-user
2. (Optional) Give the user access to the AWS Management Console. This requires a password. You must also give the user the URL of your account's sign-in page.

- aws iam create-login-profile

3. (Optional) Give the user programmatic access. This requires access keys.

- aws iam create-access-key
- Tools for Windows PowerShell: New-IAMAccessKey
- IAM API: CreateAccessKey

4. Add the user to one or more groups. The groups that you specify should have attached policies that grant the appropriate permissions for the user.

- aws iam add-user-to-group

5. (Optional) Attach a policy to the user that defines the user's permissions.

- aws iam attach-user-policy

6. (Optional) Add custom attributes to the user by attaching tags.

7. (Optional) Give the user permission to manage their own security credentials

C. Creating IAM users (AWS API)

Steps to create an IAM user from the AWS API

1. Create a user.
2. (Optional) Give the user access to the AWS Management Console. This requires a password. You must also give the user the URL of your account's sign-in page.
    - CreateLoginProfile
3. (Optional) Give the user programmatic access. This requires access keys.
    - CreateAccessKey
4. Add the user to one or more groups. The groups that you specify should have attached policies that grant the appropriate permissions for the user.
    - AddUserToGroup
5. (Optional) Attach a policy to the user that defines the user's permissions.
    - AttachUserPolicy
6. (Optional) Add custom attributes to the user by attaching tags
7. (Optional) Give the user permission to manage their own security credentials.

Migrating Application to Cloud:

- Cloud migration encompasses moving one or more enterprise applications and their IT environments from the traditional hosting type to the cloud environment, either public, private, or hybrid.
- Cloud migration presents an opportunity to significantly reduce costs incurred onapplications.
- This activity comprises, of different phases like evaluation, migration strategy, prototyping, provisioning, and testing

Phases of Cloud Migration:

1. Evaluation:

- Evaluation is carried out for all the components like current infrastructure and application architecture, environment in terms of compute, storage, monitoring, and management, SLAs, operational processes, financial considerations, risk, security, compliance, and licensing needs are identified to build a business case for moving to the cloud

2. Migration strategy:

- Based on the evaluation, a migration strategy is drawn—a hotplug strategy is used where the applications and their data and interface dependencies are isolated and these applications can be operationalized all at once.
- A fusion strategy is used where the applications can be partially migrated;
- but for a portion of it, there are dependencies based on existing licenses, specialized server requirements like mainframes, or extensive interconnections with other applications.

3. Prototyping:

- Migration activity is preceded by a prototyping activity to validate and ensure that a small portion of the applications are tested on the cloud environment with test data setup.

4. Provisioning:

- Premigration optimizations identified are implemented.
- Cloud servers are provisioned for all the identified environments, necessary platform softwares and applications are deployed, configurations are tuned to match the new environment sizing, and databases and files are replicated.
- All internal and external integration points are properly configured.
- Web services, batch jobs, and operation and management software are set up in thenew environments.

5. Testing:

- Postmigration tests are conducted to ensure that migration has been successful.
- Performance and load testing, failure and recovery testing, and scale-out testing are conducted against the expected traffic load and resource utilization levels.

Approaches for Cloud Migration:

The following are the four broad approaches for cloud migration that have been adopted effectively by vendors:

1. **Migrate existing applications**:
    - Rebuild or re architect some or all the applications, taking advantage of some of the virtualization technologies around to accelerate the work.
    - But, it requires top engineers to develop new functionality.
    - This can be achieved over the course of several releases with the timing determined by customer demand.
2. **Start from scratch:**
    - Rather than cannibalize sales, confuse customers with choice, and tie up engineers trying to rebuild existing applications. it may be easier to start again.
    - Many of the R&D decisions will be different now, and with some of the more
    - sophisticated development environments, one can achieve more even with a smallfocused working team.
3. **Separate company**:
    - One may want to create a whole new company with separate brand, management,R&D, and sales.
    - The investment and internet protocol (IP) may come from the existing company, but many of the conflicts disappear once a newborn in the cloud company is established.
    - The separate company may even be a subsidiary of the existing company.
    - What is important is that the new company can act, operate, and behave like acloud-based start-up.
4. **Buy an existing cloud vendor:**
    - For a large established vendor, buying a cloud-based competitor achieves twothings.

- Firstly, it removes a competitor
- secondly, it enables the vendor to hit the ground running in the cloud space.
- The risk of course is that the innovation, drive, and operational approach of thecloud-based company are destroyed as it is merged into the larger acquirer