Unit - 2

Q.1 1 Mark.

1) Full form : AMI
→ Amazon Machine Image

2) Full form : EBS
→ Elastic Block Store

3) Full form : MFA
→ Multi-factor Authentication

4) Full form : IAM
→ Identity and access Management

5) What is EBS Snapshot ?
→ An EBS snapshot is a point-in-time backup of an Elastic Block Store (EBS) volume. It captures the data on the volume and stores it in Amazon S3, allowing users to create new EBS volumes based on the snapshot or restore volume to a previous state.

DATE:
PAGE:

Q.2

1) (a) Explain IAM.

→ IAM stands for identity and Access Management and it's a fundamental service provided by Amazon web services for securely managing access to resource in the AWS cloud environment.

* key component :-

1) Users :- Users represent individuals or entities who interact with AWS resources. Each user has a unique identity and associated credentials.

2) Groups :- Groups are collections of IAM users. By assigning permissions to Groups you can manage access more efficiently.

3) Roles :- IAM roles are similar to users, but they are not associated with specific individuals. Instead, they are assumed by entities to temporarily obtain permissions to access resources.

4) Policies :- IAM policies are JSON documents that define permissions. They specify what actions are allowed or denied on which AWS resources.

5) Access keys :- IAM provides access keys (Access key ID and secret access key) that allow programmatic access to AWS services.

6) Multi-factor Authentication (MFA) :- IAM supports MFA, adding an extra layer of security to user sign-ins and API calls.

* Benefits :-

→ Security :-
→ Granular control :-
→ Auditing and compliance :-
→ Integration :-

Q) Explain any three features of IAM.

→ (i) Granular Permissions :-
(ii) Multi-factor Authentication (MFA) :-
(iii) Identify federation :-

(i) Granular Permissions :-

→ IAM allows precise control over resource access by granting different permissions to different users or groups for specific AWS resources. This enables organizations to implement the principle of least privilege, enhancing security by limiting access to only what's necessary

(ii) Multi-factor Authentication (MFA):

→ IAM supports MFA, requiring users to provides a second form of verification when accessing AWS resources. This adds an extra layer of security, reducing the risk of unauthorized access, even if passwords are compromised.

(iii) Identity federation:

→ IAM enables integration with existing identity systems, allowing users to access AWS resources using their existing credentials from other identity providers. This streamlines user management and enhances user experience by enabling single sign-on and centralized access control.

3) Explain policy Types in IAM.

→

(i) Identify-based policies :-

→ These policies are attached to IAM identities and grants permissions directly to those identities.

(ii) Resource-based policies :-

→ These policies are attached directly to resources such as Amazon S3 buckets or IAM roles.

(iii) Permissions boundaries :-

→ Used to set the maximum permissions that an IAM entity can be granted by identity-based policies.

(iv) Organizations SCPs :-

→ SCPs are used within AWS organizations to define the maximum permissions for account members or organizational units.

(v) Access control lists (ACLs) :-

→ ACLs are used to control access to resources across accounts by specifying which principals from other accounts can access the resources.

(vi) Session policies :-

→ Advanced session policies passed when assuming a role or a federated user using the AWS CLI or API.

4) Explain IAM Identities

→ There are several types of IAM identities.

(i) Users : There are individuals or entities that interact with the cloud resources. Users can be human users, like employee or administrators, or they can be non-human entities like applications or services.

(ii) Groups : Group are collection of users. They provide a way to manage common access permissions for multiple users collectively.

(iii) Roles : Roles define a set of permissions that an entity can assume. Roles are not tied to a specific user or group, rather, they can be assumed by any entity that needs the permissions defined in the role.