IAM Roles for Services (Easy Version)

- IAM roles let AWS services get permissions **without needing usernames/passwords or access keys**.
- Example:
    - **EC2 instance** → can read/write to **S3 bucket** without saving keys.
    - **Lambda function** → can call **other AWS services** without hardcoding keys.

IAM Security Tools (Simple Version)

- **IAM Credential Report**
    - List of all IAM users.
    - Shows if their passwords/keys are active or expired.
    - Good for checking account security.
- **IAM Access Advisor**
    - Shows what services a user can access.
    - Also shows the **last time** they used those permissions.
    - Helps remove **extra/unneeded permissions**.
- **IAM Policy Simulator**
    - Test IAM policies **before using them**.
    - See which actions are **allowed or denied**.
    - Great for fixing policy issues.

IAM Best Practices ✅

1. **Least Privilege** → Give only the permissions needed.
2. **Enable MFA** → Add extra security (password + code).

3. **Use Roles, not Users** → For apps and services, use roles instead of IAM user keys.
4. **Rotate Credentials** → Change passwords/keys often, remove unused ones.
5. **Use Managed Policies** → Start with AWS-provided policies for common tasks.

Compute Pricing (Basic Idea) 💰

- **Pay-as-you-go** → Pay only for what you use.
- Options:
    - **On-Demand** → No commitment, pay hourly/second.
    - **Reserved** → 1–3 year commitment, cheaper.
    - **Spot** → Very cheap, but can be stopped anytime.

---

b. Billing Alarms in CloudWatch ⏰

- Set an **alarm** when your AWS costs go over a limit.
- Example: Notify you if your bill goes above **$20**.

---

c. AWS Budgets 📊

- Lets you set a **custom budget** for cost or usage.
- Send **alerts** (email/SNS) when you're near or over budget.
- More flexible than simple billing alarms.