Let's go through a practical example for each of the AWS services mentioned:

1. **Compute Services:**
   - **EC2 (Elastic Compute Cloud):**
     - *Example:* Suppose you want to host a web application. You can launch an EC2 instance, install a web server (e.g., Apache or Nginx), and deploy your application code. Users can access your website by connecting to the public IP or domain associated with the EC2 instance.

   - **Lambda:**
     - *Example:* Imagine you have an S3 bucket where users upload images. You can create a Lambda function that automatically triggers whenever a new image is uploaded. The Lambda function can resize the image and store the resized version back in the S3 bucket.

2. **Storage Services:**
   - **S3 (Simple Storage Service):**
     - *Example:* You have a static website, and you use S3 to store HTML, CSS, JavaScript, and image files. Users can access your website directly from the S3 bucket, and S3 serves the content with low-latency and high durability.

   - **EBS (Elastic Block Store):**
     - *Example:* You have an EC2 instance running a database server. You attach an EBS volume to store the database files. Even if the EC2 instance is terminated, the data in the EBS volume persists and can be attached to a new instance.

3. **Database Services:**
   - **RDS (Relational Database Service):**
     - *Example:* You want to set up a MySQL database for your e-commerce website. You can use RDS to create a managed MySQL database instance, handle backups, and easily scale your database as your website grows.

   - **DynamoDB:**
     - *Example:* You are developing a mobile app, and you use DynamoDB to store user profiles and preferences. DynamoDB provides low-latency access to data, and its scalability allows your app to handle a growing user base.

4. **Networking:**
   - **VPC (Virtual Private Cloud):**
     - *Example:* You create a VPC with public and private subnets. The public subnet contains a load balancer and web servers, while the private subnet contains database servers. This architecture provides security by isolating components.

   - **Route 53:**

    - *Example:* You register a domain with Route 53 and create DNS records to map your domain to an S3 bucket for hosting a static website. Users can access your website using the custom domain.

5. **Security and Identity:**
   - **IAM (Identity and Access Management):**
     - *Example:* In an organization, you create IAM users with specific permissions. For example, a developer might have permissions to launch and terminate EC2 instances, while a database administrator has permissions to manage RDS instances.

   - **Cognito:**
     - *Example:* You use Amazon Cognito to handle user authentication for a mobile app. Users can sign up, sign in, and their authentication state is managed securely by Cognito.

6. **Machine Learning:**
   - **Sagemaker:**
     - *Example:* You have a dataset for predicting customer churn. Using Sagemaker, you can train a machine learning model, deploy it as an endpoint, and integrate the endpoint with your application for real-time predictions.

   - **Rekognition:**
     - *Example:* You build a social media application, and you use Rekognition to analyze user-uploaded images, automatically tagging and categorizing content based on detected objects or faces.

7. **Analytics:**
   - **Redshift:**
     - *Example:* Your business generates large amounts of data. You use Redshift to create a data warehouse, import data from various sources, and run complex SQL queries to analyze trends, customer behavior, and make data-driven decisions.

   - **Glue:**
     - *Example:* You have data stored in various sources like S3 and databases. With Glue, you create an ETL job to transform and consolidate this data into a Redshift data warehouse for analysis.

8. **Developer Tools:**
   - **CodeCommit:**
     - *Example:* Your development team collaborates on a project, and you use CodeCommit to host the Git repository. Developers can push and pull code, manage branches, and perform version control.

   - **CodePipeline:**

- *Example:* You set up a CI/CD pipeline using CodePipeline. When developers push code changes to CodeCommit, CodePipeline automatically builds and deploys the updated application to an EC2 instance.

9. **Management and Monitoring:**
   - **CloudWatch:**
     - *Example:* You use CloudWatch to monitor the performance of your EC2 instances. You set up alarms to notify you if CPU utilization exceeds a certain threshold, allowing you to take proactive measures.

   - **CloudFormation:**
     - *Example:* You define your infrastructure using CloudFormation templates. When you need to replicate the infrastructure or make changes, CloudFormation automates the process, ensuring consistency and reducing manual errors.

10. **Internet of Things (IoT):**
    - **IoT Core:**
      - *Example:* In a smart home scenario, IoT Core manages and connects devices such as thermostats and lights. Devices communicate with IoT Core to send data or receive commands, enabling centralized control.

    - **Greengrass:**
      - *Example:* In an industrial setting, you deploy Greengrass on edge devices (e.g., machinery controllers). Greengrass runs Lambda functions locally, enabling real-time processing and reducing latency for critical operations.

These examples illustrate how various AWS services can be applied in different scenarios, showcasing the flexibility and scalability of the AWS cloud platform.