

**Name: Khan Sumaiya**

**Roll No: A031**

### **1. Platform as a Service (PaaS)**

Platform as a Service (PaaS) is a cloud computing model that offers a complete development and deployment environment in the cloud. This model allows developers to focus on application creation without worrying about the underlying infrastructure, which includes OS, middleware, and runtime configurations. PaaS supports various development and testing capabilities, enabling users to create applications more efficiently while reducing the complexity of deployment and scaling.

#### **Key Features of PaaS:**

- Simplifies app development with built-in software stacks and services.
- Automates infrastructure provisioning, scaling, and management.
- Provides flexibility to support multiple programming languages and frameworks.

**Examples of PaaS:** AWS Elastic Beanstalk, Google App Engine, Microsoft Azure App Service.

### **2. AWS Elastic Beanstalk**

AWS Elastic Beanstalk is a PaaS offering from Amazon Web Services (AWS) designed to simplify application deployment. It enables developers to deploy and manage applications without managing the infrastructure. With Elastic Beanstalk, users can quickly deploy applications in various languages (such as Java, Python, Node.js, and Ruby) by uploading the application code. Beanstalk handles the rest—provisioning the resources, load balancing, autoscaling, and monitoring the application's health.

#### **Benefits of Elastic Beanstalk:**

- Automation: Manages underlying infrastructure, making deployment and scaling easier.
- Flexibility: Allows customization of AWS resources, providing control over EC2 instances, databases, and networking settings.
- Cost-Efficiency: Users only pay for the resources used, while Beanstalk itself is free of charge.

**Name: Khan Sumaiya**

**Roll No: A031**

### **3. Components of Elastic Beanstalk**

AWS Elastic Beanstalk consists of several key components that work together to provide a streamlined environment for application deployment:

- **Environment:** The logical construct where your application runs, containing all the necessary resources (like EC2 instances, load balancers, and databases).
- **Application:** A collection of Elastic Beanstalk environments, settings, and application versions. It represents the overall project or system you are building.
- **Environment Tiers:**
  - Web Server Tier: For handling HTTP(S) requests from clients (used for web applications).
  - Worker Tier: For applications with background processing tasks or queuing mechanisms.
- **Environment Configuration:** Configurations related to instance types, autoscaling policies, load balancing, and software settings (e.g., runtime, platform).
- **Application Versions:** Different iterations of your application's code. Each version can be deployed to different environments, allowing for testing, staging, and production control.
- **Elastic Beanstalk Command Line Interface (EB CLI):** A tool for managing applications and environments from the command line, providing ease in deployment and updates.

### **4. IAM (Identity and Access Management)**

AWS Identity and Access Management (IAM) is a web service that allows secure control over AWS resources. IAM enables users to create and manage AWS users and groups and assign permissions to control access to resources. It's essential in environments like Elastic Beanstalk for managing who can access, deploy, or make changes to applications, ensuring only authorized users can manage the environment.

#### **Key IAM Concepts in Elastic Beanstalk:**

- **IAM Users:** Individual accounts for people or applications interacting with AWS. Permissions can be assigned to control their actions within Elastic Beanstalk.
- **IAM Roles:** Used to grant permissions to applications or services. Elastic Beanstalk itself requires an IAM role to interact with resources (e.g., creating instances, accessing S3 for storage).
- **Policies:** JSON documents that define permissions. Policies are attached to IAM users, groups, or roles, determining what actions they can perform on Elastic Beanstalk resources.

**Name: Khan Sumaiya**

**Roll No: A031**

**Conclusion:**

AWS Elastic Beanstalk, as a PaaS, offers a powerful and automated solution for application deployment. By integrating with IAM, Beanstalk maintains secure access and resource management, simplifying the deployment process while ensuring control over user permissions. This integration allows developers to focus on application logic and performance rather than infrastructure, accelerating development cycles and improving efficiency.