**Auto Scaling Learning Material**

**1. Auto Scaling Service Introduction**  
Amazon **Auto Scaling** is a service that automatically adjusts the number of EC2 instances in response to application demand. It ensures your application has the right amount of capacity to handle traffic and reduces costs by scaling down when demand is low.

**Key Benefits of Auto Scaling**:

* **Cost Optimization**: Automatically scales down during low usage to save costs.
* **High Availability**: Ensures sufficient instances are available to handle traffic, minimizing downtime.
* **Flexibility**: Scales up during traffic spikes, ensuring a seamless user experience.
* **Automation**: Reduces manual intervention by automating the scaling process.

**Use Case Examples**:

* Websites or applications with unpredictable traffic patterns.
* Applications requiring 24/7 uptime with varying loads.
* Batch jobs that require increased capacity temporarily.

**2. Launch Template Overview**  
A **Launch Template** is a resource that defines the configuration settings for EC2 instances used in an Auto Scaling group. It simplifies instance creation and ensures consistency in instance configurations.

**Key Components of a Launch Template**:

* **AMI (Amazon Machine Image)**: Defines the operating system and software on the instance.
* **Instance Type**: Specifies the hardware configuration (e.g., t2.micro, m5.large).
* **Key Pair**: Used for secure SSH access to the instances.
* **Security Groups**: Define inbound and outbound traffic rules.
* **User Data**: Allows you to specify initialization scripts for your instances.

**How to Create a Launch Template**:

1. Navigate to the **EC2 Dashboard** and select **Launch Templates**.
2. Click **Create Launch Template**.
3. Specify the template name and versioning (for updates).
4. Define the **AMI**, **Instance Type**, and other configurations.
5. Save the template.

**3. Auto Scaling Group**  
An **Auto Scaling Group (ASG)** is a collection of EC2 instances managed by Auto Scaling. The ASG uses a launch template to create and terminate instances based on scaling policies.

**Key Features of Auto Scaling Groups**:

* **Minimum, Desired, and Maximum Capacity**:
  + **Minimum**: The least number of instances to keep running.
  + **Desired**: The ideal number of instances during normal operation.
  + **Maximum**: The most instances allowed during scaling up.
* **Health Checks**: Monitors instances and replaces unhealthy ones automatically.
* **Scaling Policies**: Defines when and how the group scales up or down.
* **Multi-AZ Deployment**: Distributes instances across multiple Availability Zones for fault tolerance.

**Steps to Create an Auto Scaling Group**:

1. Navigate to the **Auto Scaling Groups** section in the AWS Management Console.
2. Click **Create Auto Scaling Group**.
3. Choose the launch template and specify the instance configuration.
4. Set the minimum, desired, and maximum capacity.
5. Configure health checks, scaling policies, and load balancing (if applicable).

**4. Types of Policies**  
Scaling policies determine how and when Auto Scaling adjusts the number of instances in your group. AWS offers several policy types:

* **Target Tracking Policy**:
  + Automatically adjusts capacity to maintain a specified metric, such as CPU utilization.
  + Example: Maintain 50% CPU usage across all instances.
* **Step Scaling Policy**:
  + Scales the number of instances incrementally based on alarms triggered by CloudWatch.
  + Example: Add one instance if CPU usage exceeds 70%; remove one instance if CPU usage falls below 30%.
* **Scheduled Scaling Policy**:
  + Adds or removes instances at predefined times.
  + Example: Scale up during business hours and scale down after hours.
* **Predictive Scaling Policy**:
  + Uses machine learning to predict future traffic and scales accordingly.
  + Ideal for workloads with predictable traffic patterns.

**Key Concepts in Auto Scaling**

| **Feature** | **Description** | **Example Use Case** |
| --- | --- | --- |
| **Health Checks** | Replaces unhealthy instances automatically. | Ensures all instances are operational. |
| **Scaling Policies** | Defines the conditions under which scaling occurs. | Adjusts capacity during traffic spikes. |
| **Multi-AZ Support** | Distributes instances across multiple Availability Zones. | Increases fault tolerance and availability. |
| **Dynamic Scaling** | Adjusts the number of instances in response to demand changes. | Scales for unplanned traffic increases. |
| **Scheduled Scaling** | Scales based on predefined schedules. | E-commerce scaling during sales events. |

**Hands-On**

**In this section, students will practice Auto Scaling configuration to gain hands-on experience.**

**Exercise 1: Create a Launch Template**

1. Define a launch template with a specified AMI, instance type, and user data script.
2. Save the launch template and note its ID.

**Exercise 2: Set Up an Auto Scaling Group**

1. Create an Auto Scaling Group using the launch template.
2. Configure the minimum, desired, and maximum instance counts.
3. Distribute the instances across multiple Availability Zones.

**Exercise 3: Configure Scaling Policies**

1. Set up a **Target Tracking Policy** to maintain CPU utilization at 50%.
2. Create a **Scheduled Scaling Policy** to scale up during peak hours (e.g., 9 AM–5 PM) and scale down afterward.

**Exercise 4: Test Auto Scaling Behavior**

1. Simulate a traffic spike by increasing the load on the EC2 instances.
2. Observe how the Auto Scaling Group adds instances automatically.
3. Reduce the load and watch the instances scale down.